

DIGITAL TWIN BASED INTELLIGENT EVOLUTION METHOD AND SYSTEM FOR DISTRIBUTED COMPUTING RESOURCES

TECHNICAL FIELD

5 [0001] The present invention relates to a resource evolution technology, and specifically relates to a Digital Twin based intelligent evolution method and system for distributed computing resources.

BACKGROUND

10 [0002] Distributed computing systems have become an important infrastructure for processing large-scale data and complex applications. With the rapid development of cloud computing and edge computing, the scale and complexity of distributed computing clusters are continuously increasing, which makes a key challenge to efficiently manage and schedule computing resources. Traditional resource management methods often rely on
15 static configurations and rule-based scheduling strategies, making it difficult to adapt to dynamically changing loads and complex application requirements. In addition, traditional monitoring methods mainly focus on performance indicators of individual computing nodes, lacking global perception and collaborative optimization of the resource state of the whole cluster.

20 [0003] Lack of refined perception on resource state: in the traditional monitoring methods, it is difficult to capture tiny changes and inflection points of resource state, leading to an inability to timely identify potential performance bottlenecks and failure risks. This limits the effectiveness and flexibility of resource management and scheduling strategies.

25 [0004] Lack of modeling for dynamic association between computing nodes: in traditional resource scheduling methods, the computing nodes are often treated as isolated individuals, the mutual influence and interdependence relationship between the nodes are ignored. This leads to an underutilization of the overall computing capacity of the clusters, easily causing resource wastage and performance bottlenecks.

30 [0005] Lack of fusion on multi-time-scale resource utilization prediction: in traditional resource prediction methods, it usually focuses only on a single time scale, making it difficult to balance the accuracy of short-term predictions with the stability of long-term predictions. This results in unreliable decision basis for resource management and scheduling.

SUMMARY OF THE INVENTION

[0006] The embodiments of the present invention provide a Digital Twin based intelligent evolution method and system for distributed computing resources, aiming to solve the problems in the related technology.

[0007] In a first aspect,

[0008] the Digital Twin based intelligent evolution method for distributed computing resources provided by the embodiment of the present invention includes:

[0009] deploying an intelligent perception probe at each computing node of a distributed computing cluster, the intelligent perception probe acquiring computing resource operation data by a multi-task parallel acquisition algorithm; performing feature extraction on the computing resource operation data by a double-level attention mechanism, the first-level attention mechanism identifying resource state inflection points to generate an inflection feature sequence, and the second-level attention mechanism extracting resource fluctuation features based on the inflection feature sequence to obtain a feature tensor; inputting the feature tensor into a graph neural network to construct a computing resource Digital Twin model, the graph neural network fusing a time-order convolution layer and a spatial relationship layer to establish a dynamic node association graph among the computing nodes; based on the dynamic node association graph, computing feature importance by a hybrid entropy optimization algorithm to obtain a feature weight matrix, the feature weight matrix being used for representing multi-dimensional relevancy of the resource states;

[0010] inputting the feature weight matrix and the dynamic node association graph into a hierarchical reinforcement learning framework, learning a global resource allocation strategy based on the feature weight matrix by an improved strategy gradient algorithm in a macro level, and optimizing a local scheduling decision based on the dynamic node association graph by an improved Q learning algorithm in a micro level; and

[0011] inputting the global resource allocation strategy and the local scheduling decision into an attention-enhanced sequence prediction network to generate a resource utilization prediction sequence of a multi-scale time window; based on the resource utilization prediction sequence, designing a dynamic fusion mechanism, and in combination with long/short-term prediction results, computing prediction weights of different time scales by an improved Bayesian optimization algorithm to obtain a fusion prediction result; and constructing a multi-target optimization model according to the

fusion prediction result, setting a system performance index as an optimization target, and solving by a particle swarm optimization algorithm to obtain an optimal resource allocation solution and an optimal scheduling strategy.

[0012] The inputting the feature tensor into a graph neural network to construct a
 5 computing resource Digital Twin model, the graph neural network fusing a time-order
 convolution layer and a spatial relationship layer to establish a dynamic node association
 graph among the computing nodes; based on the dynamic node association graph,
 computing feature importance by a hybrid entropy optimization algorithm to obtain a
 feature weight matrix, the feature weight matrix being used for representing
 10 multi-dimensional relevancy of the resource states includes:

[0013] inputting the feature tensor into the graph neural network that fuses the
 time-order convolution layer and the spatial relationship layer, the time-order convolution
 layer being of a multi-scale parallel convolution structure, performing causal filling to keep
 the sequence length unchanged so as to obtain multi-scale time-order features; the spatial
 15 relationship layer adopting a dynamic attention mechanism, computing an inter-node
 attention weight based on a learnable parameter matrix, and performing adaptive
 aggregation on the node features according to the inter-node attention weight;

[0014] constructing the dynamic node association graph among the computing nodes
 based on the output of the graph neural network, computing a Pearson association
 20 coefficient matrix and a resource calling dependency matrix between the nodes by a sliding
 time window mechanism, fusing the Pearson association coefficient matrix with the
 resource calling dependency matrix by the adaptive weight coefficient to obtain a
 comprehensive intensity of association matrix, and dynamically updating the
 comprehensive intensity of association matrix based on an exponential attenuation mode so
 25 as to obtain the dynamic node association graph;

[0015] based on the dynamic node association graph, computing the feature
 importance by the hybrid entropy optimization algorithm, the hybrid entropy optimization
 algorithm computing an information entropy of a single feature through kernel density
 estimation, then computing a cross entropy between the feature and a target variable,
 30 computing conditional mutual information based on the information entropy and the cross
 entropy, and normalizing the conditional mutual information to obtain the feature weight
 matrix;

[0016] dynamically optimizing the feature weight matrix, adding a time attenuation
 factor to adjust the weight according to feature timeliness, computing a weight gradient

based on prediction error loss, and updating the weight by a gradient descent method of a momentum item, the momentum item being used for smoothing the weight updating process; and

[0017] feeding back the updated feature weight matrix to the graph neural network for
 5 guiding dynamic weighted aggregation of the node features in the spatial relationship layer, thereby realizing iterative optimization of the computing resource Digital Twin model, the feature weight matrix representing the multi-dimensional relevancy of the resource states.

[0018] The inputting the feature weight matrix and the dynamic node association graph into a hierarchical reinforcement learning framework, learning a global resource
 10 allocation strategy based on the feature weight matrix by an improved strategy gradient algorithm in a macro level, and optimizing a local scheduling decision based on the dynamic node association graph by an improved Q learning algorithm in a micro level includes:

[0019] inputting the feature weight matrix and the dynamic node association graph
 15 into the hierarchical reinforcement learning framework, the hierarchical reinforcement learning framework including the macro level and the micro level, the hierarchical reinforcement learning framework implementing strategy collaboration in the macro level and the micro level based on a hierarchical reward mechanism, and the hierarchical reward mechanism constructing a strategy learning target by a weighted combination of a global
 20 reward value and a local reward value;

[0020] learning the global resource allocation strategy based on the feature weight matrix by the improved strategy gradient algorithm in the macro level, the improved strategy gradient algorithm mapping feature weight information into resource allocation action probability by a strategy network, adding a trust domain constraint to limit strategy
 25 update step length, constructing an optimization target function by a new and old strategy ratio cutting mode, and updating strategy network parameters in combination with a time-attenuated adaptive learning rate;

[0021] optimizing the local scheduling decision based on the dynamic node association graph by the improved Q learning algorithm in the micro level, the improved Q
 30 learning algorithm constructing a dual Q network structure, including a principal value network and an auxiliary value network, extracting state features of a target node and neighbor nodes based on the dynamic node association graph, combining outputs of the two value networks by a learnable fusion coefficient, and performing sample learning by a time-order difference error based priority experience playback mechanism;

[0022] implementing collaborative decision-making in the macro level and the micro level by a bidirectional information flow mechanism, transforming the global resource allocation strategy learned in the macro level into local decision constraints for guiding action selection in the micro level, and feeding back the local state information in the micro level to the macro level after pooling operation aggregation; and

[0023] constructing a cross-level value assessment function, performing weighted combination on the global value assessment in the macro level and the local value assessment in the micro level, and performing joint optimization on the global resource allocation strategy and the local scheduling decision based on the cross-level value assessment function.

[0024] The based on the resource utilization prediction sequence, designing a dynamic fusion mechanism, and in combination with long/short-term prediction results, computing prediction weights of different time scales by an improved Bayesian optimization algorithm to obtain a fusion prediction result; and constructing a multi-target optimization model according to the fusion prediction result, setting a system performance index as an optimization target, and solving by a particle swarm optimization algorithm to obtain an optimal resource allocation solution and an optimal scheduling strategy includes:

[0025] dividing the resource utilization prediction sequence into a short-term prediction sequence and a long-term prediction sequence according to different time spans, the short-term prediction sequence containing prediction values of continuous time points, and the long-term prediction sequence containing prediction values of interval sampling;

[0026] inputting the short-term prediction sequence and the long-term prediction sequence into the improved Bayesian optimization algorithm, the improved Bayesian optimization algorithm constructing a dynamic Gaussian process kernel function containing a time-varying noise item, adaptively adjusting exploration parameters based on prediction errors, and updating Gaussian process model parameters online by the sliding time window, so as to obtain fusion weights of the short-term prediction sequence and the long-term prediction sequence;

[0027] performing weighted combination on the fusion weights with the short-term prediction sequence and the long-term prediction sequence to obtain a fusion prediction result, constructing a multi-target optimization model based on the fusion prediction result, the multi-target optimization model including a system throughput target, a resource utilization rate target and an energy consumption efficiency target, and setting a resource capacity upper limit constraint, a task completion deadline constraint and a load balance

degree constraint; and

[0028] utilizing the particle swarm algorithm to solve the multi-target optimization model, the particle swarm algorithm adaptively adjusting inertia weights through nonlinear regulation factors, adding a neighborhood optimal solution into a speed updating formula to
 5 enhance local search capability, computing individual congestions based on target spatial distances and constructing a selection mechanism, storing the non-dominated solution through an external archive set, and generating the optimal resource allocation scheme and the scheduling strategy based on the solving result of the particle swarm algorithm.

[0029] The utilizing the particle swarm algorithm to solve the multi-target
 10 optimization model, the particle swarm algorithm adaptively adjusting inertia weights through nonlinear regulation factors, adding a neighborhood optimal solution into a speed updating formula to enhance local search capability, computing individual congestions based on target spatial distances and constructing a selection mechanism, storing the non-dominated solution through an external archive set, and generating the optimal
 15 resource allocation scheme and the scheduling strategy based on the solving result of the particle swarm algorithm includes:

[0030] initializing a position vector and a speed vector of a particle swarm, designing the nonlinear regulation factors, the nonlinear regulation factors being of a composite form of a cosine function and a power function, and computing a current inertia weight
 20 according to the ratio of the current number of iterations to the maximum number of iterations;

[0031] constructing a speed updating mechanism based on the current inertia weight, computing the Euclidean distance of particles in a target space to obtain a distance matrix, determining the neighborhood range of each particle according to the distance matrix,
 25 selecting an optimal solution in the neighborhood range, computing the difference between the optimal solution and the current position to obtain a neighborhood guide vector, and combining the neighborhood guide vector with an individual optimal position item and a global optimal position item to obtain a complete speed updating expression;

[0032] updating the particle position by the speed updating expression to obtain a
 30 candidate solution set, computing the function value difference of adjacent solutions in the candidate solution set in the target space to obtain a congestion index, performing non-dominated sorting on the candidate solution set according to the congestion index, and selecting the non-dominated solution to construct the external archive set;

[0033] adding the newly generated non-dominated solution into the external archive

set and re-sorting, when the external archive set exceeds a preset capacity, sorting archive members in a descending way based on the congestion index and reserving the first 5% of solutions, and storing high-quality solutions with convergence and diversity through the external archive set; and

5 [0034] computing a comprehensive assessment value for the solutions in the external archive set, selecting the solution with the optimal comprehensive assessment value as a final solution, generating a resource allocation matrix and a task scheduling sequence according to a decision variable corresponding to the final solution, and determining the optimal resource allocation scheme and the scheduling strategy based on the resource
10 allocation matrix and the task scheduling sequence.

[0035] The method further includes:

[0036] based on the optimal resource configuration scheme and the scheduling strategy, executing dynamic resource adjustment by a distributed feedback controller, the distributed feedback controller implementing accurate control by a sliding mode control
15 algorithm; constructing a multi-level anomaly detection framework to monitor an execution effect in real time, taking the feature weight matrix as a reference to identify state deviation in combination with an improved isolated forest algorithm, and analyzing a deviation transmission path based on the dynamic node association graph;

[0037] constructing a system control target based on the optimal resource
20 configuration scheme and the scheduling strategy, transforming the system control target into a state space expression, designing a sliding mode switching function of the distributed feedback controller according to the state space expression, constructing a composite control law of an equivalent control item and a switching control item based on the sliding mode switching function, and generating a dynamic resource adjustment instruction by
25 boundary layer smoothing processing;

[0038] executing the dynamic resource adjustment instruction and collecting real-time state data, constructing a multi-level anomaly detection framework containing the state features, weight matrix and detection threshold, comparing the real-time state data with a reference state represented by the feature weight matrix, and computing to obtain a state
30 deviation vector;

[0039] inputting the state deviation vector into the improved isolated forest algorithm, setting a dynamic splitting probability and an adaptive depth limit for the improved isolated forest algorithm according to the importance degree of resource features, fusing detection results of a plurality of isolated trees in a weighted voting mode, and outputting position

information and degree information of an abnormal state; and

[0040] positioning an abnormal source node in the dynamic node association graph based on the position information, computing a product of the feature weight and intensity of association between adjacent nodes according to the degree information to obtain the transmission intensity of the state deviation, analyzing a diffusion path of the transmission intensity by the dynamic node association graph, and determining a key influence link.

[0041] The method further includes:

[0042] optimizing a Digital Twin model based on the updated feature weight matrix and the dynamic association graph by an improved meta learning algorithm, and implementing quick iteration of the model through task decomposition and gradient accumulation; finally, analyzing a performance degradation root cause based on an improved causal inference model, and feeding back the analysis result to the hierarchical reinforcement learning framework to realize dynamic optimization on the decision strategy;

[0043] initializing the feature weight matrix by the improved meta learning algorithm, each element in the feature weight matrix representing an influence weight of an input feature on an output index, acquiring a task-specific weight matrix by computing a loss function of each task, computing a meta gradient based on the task-specific weight matrix, and updating a global weight matrix;

[0044] constructing the dynamic association graph based on the global weight matrix, nodes in the dynamic association graph representing system components, edges representing intensity of association among the components, screening a significant association relationship by setting a dynamic threshold, and continuously updating the intensity of association by the sliding time window;

[0045] decomposing the optimization target into a plurality of subtasks and computing gradients in parallel, allocating a weight coefficient to each subtask according to the association degree of the components in the dynamic association graph, and updating parameters of the Digital Twin model by a weighted gradient accumulation mode; and

[0046] initializing a causal graph structure based on the dynamic association graph, optimizing a causal relationship by a bidirectional search algorithm and eliminating false association, quantifying the contribution degree of an influence factor to performance degradation by inverse factual inference, inputting the contribution degree as a reward signal into the hierarchical reinforcement learning framework, the hierarchical reinforcement learning framework including the macro level used for selecting macro actions and a micro level used for executing specific operations, and implementing

collaborative optimization of the macro level and the micro level through the hierarchical reward mechanism.

[0047] In a second aspect,

[0048] the Digital Twin based intelligent evolution system for distributed computing
5 resources provided by the embodiment of the present invention includes:

[0049] a first unit, configured to deploy an intelligent perception probe at each
computing node of a distributed computing cluster, the intelligent perception probe
acquiring computing resource operation data by a multi-task parallel acquisition algorithm;
perform feature extraction on the computing resource operation data by a double-level
10 attention mechanism, the first-level attention mechanism identifying resource state
inflection points to generate an inflection feature sequence, and the second-level attention
mechanism extracting resource fluctuation features based on the inflection feature sequence
to obtain a feature tensor; input the feature tensor into a graph neural network to construct a
computing resource Digital Twin model, the graph neural network fusing a time-order
15 convolution layer and a spatial relationship layer to establish a dynamic node association
graph among the computing nodes; based on the dynamic node association graph, compute
feature importance by a hybrid entropy optimization algorithm to obtain a feature weight
matrix, the feature weight matrix being used for representing multi-dimensional relevancy
of the resource states;

20 [0050] a second unit, configured to input the feature weight matrix and the dynamic
node association graph into a hierarchical reinforcement learning framework, learn a global
resource allocation strategy based on the feature weight matrix by an improved strategy
gradient algorithm in a macro level, and optimize a local scheduling decision based on the
dynamic node association graph by an improved Q learning algorithm in a micro level; and

25 [0051] a third unit, configured to input the global resource allocation strategy and the
local scheduling decision into an attention-enhanced sequence prediction network to
generate a resource utilization prediction sequence of a multi-scale time window; based on
the resource utilization prediction sequence, design a dynamic fusion mechanism, and in
combination with long/short-term prediction results, compute prediction weights of
30 different time scales by an improved Bayesian optimization algorithm to obtain a fusion
prediction result; and construct a multi-target optimization model according to the fusion
prediction result, set a system performance index as an optimization target, and solve by a
particle swarm optimization algorithm to obtain an optimal resource allocation solution and
an optimal scheduling strategy.

[0052] In a third aspect,

[0053] an electronic device provided by the embodiment of the present invention includes:

[0054] a processor; and

5 [0055] a memory configured to store processor executable instructions; and

[0056] the processor is configured to call instructions stored in the memory to execute the abovementioned method.

[0057] In a fourth aspect,

10 [0058] the embodiment of the present invention provides a computer-readable storage medium having computer program instructions stored therein, when executed by a processor, the computer program instructions execute the abovementioned method.

[0059] The present invention has the following beneficial effects:

15 [0060] 1. the resource perception and modeling precision is improved: the multi-task parallel acquisition algorithm and the double-level attention mechanism are adopted, which can perceive and extract more accurately the key features in computing resource operation data, and in combination with the graph neural network and the time-order convolution, the more accurate computing resource Digital Twin model can be constructed, and thus the dynamic association between the resource states and the nodes can be reflected more accurately.

20 [0061] 2. The resource allocation and scheduling efficiency are optimized: based on the hierarchical reinforcement learning framework, and in combination with the global resource allocation strategy and the local scheduling decision, the computing resources can be utilized more effectively, thereby improving the resource utilization rate and the scheduling efficiency. Moreover, the application of the hybrid entropy optimization
25 algorithm can better capture the multi-dimensional relevancy of the resource states, so that the intelligent level of the decision is further improved.

[0062] 3. The resource prediction and configuration capability are enhanced: the attention-enhanced sequence prediction network and the dynamic fusion mechanism can generate a more accurate resource utilization prediction sequence of the multi-scale time
30 window, thus providing a reliable basis for resource configuration. In combination with the multi-target optimization model and the particle swarm algorithm, a better resource configuration scheme and a better scheduling strategy can be obtained, so that the overall performance of the system is improved.

BRIEF DESCRIPTION OF THE DRAWINGS

[0063] FIG. 1 is a flowchart of a Digital Twin based intelligent evolution method for distributed computing resources provided by an embodiment of the present invention; and

[0064] FIG. 2 is a flowchart of a Digital Twin based intelligent evolution system for distributed computing resources provided by an embodiment of the present invention.

DETAILED DESCRIPTION

[0065] In order to make the purpose, technical solution and advantages of the embodiment of the present invention clearer, the technical solutions in the embodiments of the present invention will be clearly and completely described in combination with the accompanying drawings in the embodiments of the present invention. Definitely, the described embodiments are only a part of the embodiments of the present invention, not all of them. Based on the embodiments in the present invention, all other embodiments obtained by those of ordinary skill in the art without carrying out creative labor fall within the scope of protection of the present invention.

[0066] The technical solutions of the present invention are described in detail by specific embodiments below. The following specific embodiments can be combined with each other, and the same or similar concepts or processes may not be repeated in some embodiments.

[0067] FIG. 1 is a flowchart of a Digital Twin based intelligent evolution method for distributed computing resources provided by an embodiment of the present invention, as shown in FIG. 1, the method includes:

[0068] S11: deploying an intelligent perception probe at each computing node of a distributed computing cluster, the intelligent perception probe acquiring computing resource operation data by a multi-task parallel acquisition algorithm; performing feature extraction on the computing resource operation data by a double-level attention mechanism, the first-level attention mechanism identifying resource state inflection points to generate an inflection feature sequence, and the second-level attention mechanism extracting resource fluctuation features based on the inflection feature sequence to obtain a feature tensor; inputting the feature tensor into a graph neural network to construct a computing resource Digital Twin model, the graph neural network fusing a time-order convolution layer and a spatial relationship layer to establish a dynamic node association graph among the computing nodes; based on the dynamic node association graph, computing feature importance by a hybrid entropy optimization algorithm to obtain a feature weight matrix,

the feature weight matrix being used for representing multi-dimensional relevancy of the resource states;

[0069] S12: inputting the feature weight matrix and the dynamic node association graph into a hierarchical reinforcement learning framework, learning a global resource allocation strategy based on the feature weight matrix by an improved strategy gradient algorithm in a macro level, and optimizing a local scheduling decision based on the dynamic node association graph by an improved Q learning algorithm in a micro level; and

[0070] S13: inputting the global resource allocation strategy and the local scheduling decision into an attention-enhanced sequence prediction network to generate a resource utilization prediction sequence of a multi-scale time window; based on the resource utilization prediction sequence, designing a dynamic fusion mechanism, and in combination with long/short-term prediction results, computing prediction weights of different time scales by an improved Bayesian optimization algorithm to obtain a fusion prediction result; and constructing a multi-target optimization model according to the fusion prediction result, setting a system performance index as an optimization target, and solving by a particle swarm optimization algorithm to obtain an optimal resource allocation solution and an optimal scheduling strategy.

[0071] In an optional embodiment, the inputting the feature tensor into a graph neural network to construct a computing resource Digital Twin model, the graph neural network fusing a time-order convolution layer and a spatial relationship layer to establish a dynamic node association graph among the computing nodes; based on the dynamic node association graph, computing feature importance by a hybrid entropy optimization algorithm to obtain a feature weight matrix, the feature weight matrix being used for representing multi-dimensional relevancy of the resource states includes:

[0072] inputting the feature tensor into the graph neural network that fuses the time-order convolution layer and the spatial relationship layer, the time-order convolution layer being of a multi-scale parallel convolution structure, performing causal filling to keep the sequence length unchanged so as to obtain multi-scale time-order features; the spatial relationship layer adopting a dynamic attention mechanism, computing an inter-node attention weight based on a learnable parameter matrix, and performing adaptive aggregation on the node features according to the inter-node attention weight;

[0073] constructing the dynamic node association graph among the computing nodes based on the output of the graph neural network, computing a Pearson association coefficient matrix and a resource calling dependency matrix between the nodes by a sliding

time window mechanism, fusing the Pearson association coefficient matrix with the resource calling dependency matrix by the adaptive weight coefficient to obtain a comprehensive intensity of association matrix, and dynamically updating the comprehensive intensity of association matrix based on an exponential attenuation mode so as to obtain the dynamic node association graph;

[0074] based on the dynamic node association graph, computing the feature importance by the hybrid entropy optimization algorithm, the hybrid entropy optimization algorithm computing an information entropy of a single feature through kernel density estimation, then computing a cross entropy between the feature and a target variable, computing conditional mutual information based on the information entropy and the cross entropy, and normalizing the conditional mutual information to obtain the feature weight matrix;

[0075] dynamically optimizing the feature weight matrix, adding a time attenuation factor to adjust the weight according to feature timeliness, computing a weight gradient based on prediction error loss, and updating the weight by a gradient descent method of a momentum item, the momentum item being used for smoothing the weight updating process; and

[0076] feeding back the updated feature weight matrix to the graph neural network for guiding dynamic weighted aggregation of the node features in the spatial relationship layer, thereby realizing iterative optimization of the computing resource Digital Twin model, the feature weight matrix representing the multi-dimensional relevancy of the resource states.

[0077] According to a computing resource Digital Twin model construction method, time-order and spatial features of the computing resources are fused by the graph neural network, the feature weights are dynamically adjusted by the hybrid entropy optimization algorithm, and thus accurate representation of the state of the computing resources is realized.

[0078] Firstly, various performance index data of the computing resources is collected, such as CPU utilization rate, internal memory utilization rate, network flow, and disk IO, and such data is constructed into the feature tensor. The resource index of each time point forms one feature vector, and the feature vectors of a plurality of time points are arranged according to a time order to form the feature tensor. For example, four indexes, namely, the CPU utilization rate, the internal memory utilization rate, the network flow and the disk IO, are collected once every 5 seconds, and the data of 100 time points are continuously collected to construct a 100 x 4 feature tensor.

[0079] Then, the feature tensor is inputted into a graph neural network that fuses the time-order convolution layer and the spatial relationship layer. In the time-order convolution layer, it extracts the time-order features by a multi-scale parallel convolution structure. For example, three parallel convolution kernels are used and have the dimension of 3, 5, and 7 respectively, and convolution operation is carried out on the inputted feature tensor. In order to keep the sequence length unchanged, the causal filling is performed, namely 0 is filled at the beginning of the sequence. By multi-scale convolution, the time-order features of different time granularities can be captured. In the spatial relationship layer, it performs adaptive aggregation on the node features by the dynamic attention mechanism. If there are three computing nodes, in the spatial relationship layer, it will dynamically compute the attention weights according to the relevancy between the nodes. For example, if the relevancy between a node 1 and a node 2 is high, the attention weight between the node 1 and the node 2 will be high, and when performing feature aggregation, the features of the node 1 and the node 2 will be endowed with larger weights.

[0080] The output of the graph neural network is used for constructing the dynamic node association graph among the computing nodes. The sliding time window mechanism is utilized, for example, resource data of past 10 time points are selected, the Pearson association coefficient matrix and the resource calling dependency matrix among the nodes are computed. The Pearson association coefficient matrix reflects the linear relevancy of resource index changes among the nodes. The resource calling dependency matrix represents a calling relationship among the nodes, for example, times of calling the node 2 by the node 1. Then, the Pearson association coefficient matrix and the resource calling dependency matrix are fused through the adaptive weight coefficient, so as to obtain the comprehensive intensity of association matrix. Finally, the comprehensive intensity of association matrix is dynamically updated based on the exponential attenuation mode so as to obtain the dynamic node association graph. The exponential attenuation mode means that the influence of recent data on the intensity of association is greater.

[0081] Based on the dynamic node association graph, the feature importance is computed by the hybrid entropy optimization algorithm. Firstly, the information entropy of the single feature is computed by kernel density estimation, and the higher the information entropy is, the larger the information amount contained in the feature is. Secondly, the cross entropy between the feature and the target variable (for example, the overall performance of the system) is computed, and the lower the cross entropy is, the higher the relevancy between the feature and the target variable is. Thirdly, the conditional mutual information is

computed based on the information entropy and the cross entropy, and the conditional mutual information represents the degree of uncertainty reduction of the target variable under the condition that the feature is known. Fourthly, the conditional mutual information is normalized to obtain the feature weight matrix. For example, if the conditional mutual information of the CPU utilization rate is the highest, the corresponding weight value in the feature weight matrix is the maximum.

[0082] The feature weight matrix is dynamically optimized. The time attenuation factor is added, and the weight is adjusted according to the feature timeliness, for example, if the influence of the CPU utilization rate on the system performance is large in the recent period of time, the weight will be higher. The weight gradient is computed based on the prediction error loss, and the weight is updated by the gradient descent method with the momentum item. The momentum item can smooth the weight updating process, thus avoiding oscillation. The updated feature weight matrix is fed back to the graph neural network for guiding dynamic weighted aggregation of the node features in the spatial relationship layer, thereby realizing iterative optimization of the computing resource Digital Twin model.

[0083] Through the solution of the present invention:

[0084] the model precision can be improved: by fusing the time-order and spatial features and dynamically adjusting the feature weight, the multi-dimensional relevancy of the computing resource state can be reflected more accurately, and therefore the precision of the Digital Twin model is improved. The model adaptability is enhanced: the dynamic attention mechanism and the time attenuation factors are adopted, so that the model can adapt to the dynamic change of the computing resource state, thereby improving the robustness and the generalization capability of the model. The resource management is optimized: an accurate Digital Twin model is constructed so that the operational state of the computing resources can be better understood, the resource scheduling and management strategies are optimized, and the resource utilization efficiency is improved.

[0085] In an optional embodiment, inputting the feature weight matrix and the dynamic node association graph into a hierarchical reinforcement learning framework, learning a global resource allocation strategy based on the feature weight matrix by an improved strategy gradient algorithm in a macro level, and optimizing a local scheduling decision based on the dynamic node association graph by an improved Q learning algorithm in a micro level includes:

[0086] inputting the feature weight matrix and the dynamic node association graph

into the hierarchical reinforcement learning framework, the hierarchical reinforcement learning framework including the macro level and the micro level, the hierarchical reinforcement learning framework implementing strategy collaboration in the macro level and the micro level based on a hierarchical reward mechanism, and the hierarchical reward mechanism constructing a strategy learning target by a weighted combination of a global reward value and a local reward value;

[0087] learning the global resource allocation strategy based on the feature weight matrix by the improved strategy gradient algorithm in the macro level, the improved strategy gradient algorithm mapping feature weight information into resource allocation action probability by a strategy network, adding a trust domain constraint to limit strategy update step length, constructing an optimization target function by a new and old strategy ratio cutting mode, and updating strategy network parameters in combination with a time-attenuated adaptive learning rate;

[0088] optimizing the local scheduling decision based on the dynamic node association graph by the improved Q learning algorithm in the micro level, the improved Q learning algorithm constructing a dual Q network structure, including a principal value network and an auxiliary value network, extracting state features of a target node and neighbor nodes based on the dynamic node association graph, combining outputs of the two value networks by a learnable fusion coefficient, and performing sample learning by a time-order difference error based priority experience playback mechanism;

[0089] implementing the collaborative decision-making in the macro level and the micro level by a bidirectional information flow mechanism, transforming the global resource allocation strategy learned in the macro level into local decision constraints for guiding action selection in the micro level, and feeding back the local state information in the micro level to the macro level after pooling operation aggregation; and

[0090] constructing a cross-level value assessment function, performing weighted combination on the global value assessment in the macro level and the local value assessment in the micro level, and performing joint optimization on the global resource allocation strategy and the local scheduling decision based on the cross-level value assessment function.

[0091] A resource scheduling method based on hierarchical reinforcement learning is used for optimizing the resource allocation and task scheduling in a complex system. According to the method, global resource allocation and local task scheduling are decoupled, and the improved reinforcement learning algorithm is adopted for collaborative

optimization in the macro level and the micro level respectively.

[0092] Firstly, a hierarchical reinforcement learning framework is constructed, and the framework contains the macro level and the micro level. The macro level is used for global resource allocation, and the micro level is used for local task scheduling. The core of the framework is the hierarchical reward mechanism in which global rewards and local rewards are subjected to weighted combination to construct a strategy learning target. The global rewards balances the overall system performance, such as throughput or delay; and the local rewards balances the performance of a single node, such as task completion time or energy consumption. For example, the global rewards can be set to be the total time for completing all tasks, and the local rewards can be set to be the time for completing the tasks on each node. The weighting coefficient can be adjusted according to system requirements, for example, if more attention is paid to overall performance, the weight of the global rewards can be set to be higher.

[0093] In the macro level, the improved strategy gradient algorithm is adopted to learn the global resource allocation strategy. The algorithm uses a strategy network to map the feature weight information into the resource allocation action probability. The feature weight matrix describes the importance of different resource types to different tasks. For example, one row in the matrix represents one resource type, one column represents one task type, and each element represents the importance of the resource type to the task type. The input of the strategy network refers to the feature weight matrix, and the output of the strategy network refers to probability of each resource allocation scheme. In order to stabilize a training process, trust domain constraints are added to limit the strategy update step length, and the new and old strategy ratio cutting mode is adopted to construct the optimal target function. In addition, a time-attenuated adaptive learning rate is adopted to update strategy network parameters, so as to improve the learning efficiency. For example, an initial learning rate is set to be 0.01 and gradually decreases along with increase of the number of training steps.

[0094] In the micro level, the improved Q learning algorithm is adopted to optimize the local scheduling decision. The algorithm constructs the dual Q network structure, including the principal value network and the auxiliary value network. Both the networks are based on the dynamic node association graphs to extract the state features of the target node and the neighbor nodes. The dynamic node association graph describes a connection relationship and state information between the nodes. For example, each node in the graph represents one computing node, edges represent communication links between the nodes,

and the features of the nodes can include the CPU utilization rate, the internal memory occupancy rate and the like. The outputs of the two value networks are combined through the learnable fusion coefficient, thus obtaining more accurate value estimation. For example, the fusion coefficient can be initialized to be 0.5 and adjusted along with the training process. In addition, a priority experience playback mechanism based on the time-order difference error is adopted to perform sample learning, thereby improving the learning efficiency. For example, a sample with a relatively large time-order difference error is endowed with a higher priority for faster learning.

[0095] In order to realize the collaborative decision-making in the macro level and the micro level, the bidirectional information flow mechanism is adopted. The global resource allocation strategy learned in the macro level is transformed into the local decision constraints for guiding the action selection in the micro level. For example, the total amount of resources allocated to a certain node in the macro level can be used as the upper limit of the number of scheduling tasks of the node. Moreover, the local state information in the micro level is aggregated by pooling operation and then fed back to the macro level, so that it can better perceive the system state in the macro level. For example, the CPU utilization average value of each node can be used as the input feature in the macro level.

[0096] Finally, the cross-level value assessment function is constructed, and the global value assessment in the macro level and the local value assessment in the micro level are subjected to weighted combination. The joint optimization is carried out on the global resource allocation strategy and the local scheduling decision based on the cross-level value assessment function, so as to realize the optimal overall system performance.

[0097] Through the solution of the present invention:

[0098] the resource utilization rate can be improved: by collaborative optimization of the global resource allocation and the local task scheduling, the system resources can be more effectively utilized, thus avoiding resource waste and bottlenecks. The system performance can be improved: the hierarchical reinforcement learning framework can dynamically adjust the resource allocation and task scheduling strategies according to the system states, so that the overall system performance can be improved, such as throughput and delay. The system adaptability can be enhanced: the method can adapt to dynamically changing environments and task loads, and automatically learn the optimal resource scheduling strategy, thereby improving the robustness and adaptability of the system.

[0099] In an optional embodiment, the based on the resource utilization prediction sequence, designing a dynamic fusion mechanism, and in combination with long/short-term

prediction results, computing prediction weights of different time scales by an improved Bayesian optimization algorithm to obtain a fusion prediction result; and constructing a multi-target optimization model according to the fusion prediction result, setting a system performance index as an optimization target, and solving by a particle swarm optimization
 5 algorithm to obtain an optimal resource allocation solution and an optimal scheduling strategy includes:

[0100] dividing the resource utilization prediction sequence into a short-term prediction sequence and a long-term prediction sequence according to different time spans, the short-term prediction sequence containing prediction values of continuous time points,
 10 and the long-term prediction sequence containing prediction values of interval sampling;

[0101] inputting the short-term prediction sequence and the long-term prediction sequence into the improved Bayesian optimization algorithm, the improved Bayesian optimization algorithm constructing a dynamic Gaussian process kernel function containing a time-varying noise item, adaptively adjusting exploration parameters based on prediction
 15 errors, and updating Gaussian process model parameters online by the sliding time window, so as to obtain fusion weights of the short-term prediction sequence and the long-term prediction sequence;

[0102] performing weighted combination on the fusion weights with the short-term prediction sequence and the long-term prediction sequence to obtain a fusion prediction
 20 result, constructing a multi-target optimization model based on the fusion prediction result, the multi-target optimization model including a system throughput target, a resource utilization rate target and an energy consumption efficiency target, and setting a resource capacity upper limit constraint, a task completion deadline constraint and a load balance degree constraint; and

[0103] utilizing the particle swarm algorithm to solve the multi-target optimization model, the particle swarm algorithm adaptively adjusting inertia weights through nonlinear regulation factors, adding a neighborhood optimal solution into a speed updating formula to enhance local search capability, computing individual congestions based on target spatial distances and constructing a selection mechanism, storing non-dominated solution through
 25 an external archive set, and generating the optimal resource allocation scheme and the scheduling strategy based on the solving result of the particle swarm algorithm.

[0104] In this embodiment, the resource utilization prediction sequence is divided into the short-term prediction sequence and the long-term prediction sequence according to different time spans. The short-term prediction sequence contains the prediction values of

the continuous time points and is used for capturing instantaneous fluctuation and short-term trend of resource utilization, and the time span is generally 5 min to 1 hour; and the long-term prediction sequence contains the prediction values of interval sampling and is used for obtaining a long-term change mode of resource utilization, and the time span is generally 6 hours to 24 hours. For example, for the CPU utilization rate of a certain cloud computing cluster, the short-term prediction sequence may contain the prediction values of consecutive 5 min such as {45.2%, 47.8%, 46.5%, 48.9%}, and the long-term prediction sequence may contain the prediction values of every 6 h such as {51.3%, 62.7%, 57.4%, 49.8%}.

[0105] The short-term prediction sequence and the long-term prediction sequence are inputted into the improved Bayesian optimization algorithm to determine the fusion weights. In the improved Bayesian optimization algorithm, it constructs the dynamic Gaussian process kernel function containing the time-varying noise item. Specifically, the kernel function is in a form of a combination of a radial basis function and a periodic function, and the time-varying noise item is added to adapt to the dynamic change characteristic of resource load. For example, for a certain moment t and a certain moment t' , the kernel function can be represented as a measure of the similarity of sample points of the two moments, containing a basic similarity item and a noise item changing along with time. In this algorithm, it adaptively adjusts the exploration parameters based on the prediction errors, and when the error of short-term or long-term prediction increases, the exploration intensity will be correspondingly increased to find preferred fusion weights; and when the prediction errors are stable or decrease, the exploration intensity will be reduced to finely adjust the existing weights.

[0106] In the improved Bayesian optimization algorithm, it updates Gaussian process model parameters online by the sliding time window. The size of the time window is set to be 120 sampling points, and 30 sampling points are updated in each window sliding, thereby making sure that the model parameters can reflect the current system state in time without being excessively influenced by past data. The updating process includes: re-computing the kernel function parameters and noise variance, and updating the mean value and variance of posterior distribution. Through this method, the algorithm can adapt to the dynamic change of the resource utilization mode, and proper fusion weights can be allocated to the short-term prediction sequence and the long-term prediction sequence.

[0107] In a specific embodiment, the weight of the short-term prediction sequence dynamically changes between 0.65 and 0.85 along with time, and the weight of the

long-term prediction sequence correspondingly changes between 0.35 and 0.15. When the system load suddenly changes, the weight of the short-term prediction sequence will automatically increase to better respond to the instant change; and in the period that the system load is relatively stable, the weight of the long-term prediction sequence will correspondingly increase to provide a more stable prediction result.

[0108] The fusion weights are subjected to weighted combination with the short-term prediction sequence and the long-term prediction sequence to obtain the fusion prediction result. Specifically, for each prediction time point t , the fusion prediction value is equal to the sum of a product obtained by multiplying the short-term prediction value by the short-term weight and a product obtained multiplying the long-term prediction value by the long-term weight. For example, if the short-term prediction value of a certain moment is 48.9%, the long-term prediction value is 51.3%, and the corresponding weights are 0.7 and 0.3 respectively, the fusion prediction value will be $48.9\% \times 0.7 + 51.3\% \times 0.3 = 49.62\%$. According to this weighted fusion method, the instantaneity of short-term prediction and the stability of long-term prediction are fully utilized, thereby improving the overall prediction accuracy.

[0109] The multi-target optimization model is constructed based on the fusion prediction result, and the model includes the system throughput target, the resource utilization rate target and the energy consumption efficiency target. The system throughput target aims at maximizing the number of tasks completed in unit time, the resource utilization rate target aims at improving the effective utilization degree of various resources, and the energy consumption efficiency target aims at reducing the energy consumption of unit computing tasks. In this model, the resource capacity upper limit constraint, the task completion deadline constraint and the load balance degree constraint are set at the same time, so as to ensure the feasibility of the optimization scheme and the stable operation of the system.

[0110] The resource capacity upper limit constraint ensures that the total number of resources allocated to each task does not exceed the available resources of the system. For example, for a cluster containing 100 computing nodes, the total number of nodes allocated at any moment does not exceed 100. The task completion deadline constraint ensures that the task can be completed before the specified deadline, thus preventing the service level protocol from being violated. The load balance degree constraint requires that the load difference of each node is controlled within a reasonable range, for example, the ratio of the maximum load to the minimum load does not exceed 1.5, so that the performance

bottleneck caused by local overload is avoided.

[0111] The multi-target optimization model is solved through the particle swarm algorithm. In the particle swarm algorithm, it adaptively adjusts the inertia weight through the nonlinear adjustment factors, and thus the inertia weight dynamically changes along with the iteration process. In the initial stage, the inertia weight is large and is about 0.9, which is conducive to global search; and as iteration goes on, the inertia weight gradually decreases to be about 0.4, and thus the local search capability is enhanced. In the algorithm, the neighborhood optimal solution is introduced into the speed update formula, the neighborhood is defined as a particle set with the Euclidean distance less than the preset threshold (such as 0.2), and the added neighborhood optimal solution improves the local search efficiency of the algorithm.

[0112] In the particle swarm algorithm, it computes the individual congestions based on the target space distance and constructs the selection mechanism. The congestions represent the density degree of the particles in the target space, the selected probability of the particles in the area with high congestion is decreased, and the selected probability of the particles in the area with low congestion is increased, and therefore, the diversity of a solution set is kept. In the algorithm, the non-dominated solutions are stored through the external archive set, the capacity of the archive set is set to be 100, and when the number of the non-dominated solutions exceeds capacity limitation, the solution with high diversity will be selected and reserved based on the congestion.

[0113] The optimal resource allocation scheme and the scheduling strategy are generated based on the solution result of the particle swarm algorithm. The optimal resource allocation scheme indicates the allocation number of various resources (such as CPU, internal memory and network bandwidth) in different time periods, and the scheduling strategy includes a task allocation sequence, an execution priority and a dynamic resource adjustment rule. For example, in a certain business scenario, the optimized resource allocation scheme shows that: in a peak period (such as 10:00 a.m. to 12:00 a.m. and 2:00 p.m. to 4:00 p.m. every day), the CPU core allocation is increased by 30% and the internal memory allocation is increased by 20%; in a trough period (such as 11:00 p.m. to 05:00 a.m. the next day), the CPU core allocation and internal memory allocation are decreased by 40% and 25% respectively, meanwhile, non-critical tasks are scheduled to the trough period to be executed, and high-performance resources are preferentially allocated to critical tasks. By means of such resource allocation and scheduling strategy, the system throughput is increased by 25%, the average resource

utilization rate is increased by 18%, and the energy consumption efficiency is improved by 22%.

[0114] In an optional embodiment, the utilizing the particle swarm algorithm to solve the multi-target optimization model, the particle swarm algorithm adaptively adjusting inertia weights through nonlinear regulation factors, adding a neighborhood optimal solution into a speed updating formula to enhance local search capability, computing individual congestions based on target spatial distances and constructing a selection mechanism, storing non-dominated solution through an external archive set, and generating the optimal resource allocation scheme and the scheduling strategy based on the solving result of the particle swarm algorithm includes:

[0115] initializing a position vector and a speed vector of a particle swarm, designing the nonlinear regulation factors, the nonlinear regulation factors being of a composite form of a cosine function and a power function, and computing a current inertia weight according to the ratio of the current number of iterations to the maximum number of iterations;

[0116] constructing a speed updating mechanism based on the current inertia weight, computing the Euclidean distance of particles in a target space to obtain a distance matrix, determining the neighborhood range of each particle according to the distance matrix, selecting an optimal solution in the neighborhood range, computing the difference between the optimal solution and the current position to obtain a neighborhood guide vector, and combining the neighborhood guide vector with an individual optimal position item and a global optimal position item to obtain a complete speed updating expression;

[0117] updating the particle position by the speed updating expression to obtain a candidate solution set, computing the function value difference of adjacent solutions in the candidate solution set in the target space to obtain a congestion index, performing non-dominated sorting on the candidate solution set according to the congestion index, and selecting a non-dominated solution to construct the external archive set;

[0118] adding the newly generated non-dominated solution into the external archive set and re-sorting, when the external archive set exceeds a preset capacity, sorting archive members in a descending way based on the congestion index and reserving the first 5% of solutions, and storing high-quality solutions with convergence and diversity through the external archive set; and

[0119] computing a comprehensive assessment value for the solutions in the external archive set, selecting the solution with the optimal comprehensive assessment value as a

final solution, generating a resource allocation matrix and a task scheduling sequence according to a decision variable corresponding to the final solution, and determining the optimal resource allocation scheme and the scheduling strategy based on the resource allocation matrix and the task scheduling sequence.

5 [0120] A multi-target resource allocation and scheduling method based on an improved particle swarm algorithm is mainly used for improving the resource utilization rate and optimizing the scheduling efficiency. According to the method, by nonlinear adjustment of the inertia weights, addition of the neighborhood optimal solution, construction of a selection mechanism based on the congestion, maintenance of the external
10 archive set and other strategies, the search capability and convergence of the algorithm are enhanced, and finally the optimal resource allocation scheme and scheduling strategy are generated.

 [0121] Firstly, the particle swarm is initialized. Each particle represents a resource allocation and scheduling scheme, and the position vector of each particle represents
15 decision variables in the scheme, for example, the resource allocation proportion, and the task execution sequence. The speed vector represents the moving direction and speed of each particle in the search space. The initial position and speed are randomly generated, so that the diversity of the swarm is ensured. For example, if there are 10 resources and 5 tasks, the position vector of each particle contains 15 elements, representing which task
20 each resource is allocated to and the execution sequence of the tasks respectively. The initial speed vector also contains 15 elements, and the value range is randomly generated from -1 to 1.

 [0122] Then, the nonlinear regulation factors are designed to dynamically adjust the inertia weights. The inertia weights influence the degree of the particles inheriting the
25 previous speed. A composite form of the cosine function and the power function is adopted, and the current inertia weight is computed according to the ratio of the current iteration number to the maximum iteration number. For example, if the maximum number of iterations is 100 and the current number of iterations is 50, the current inertia weight can be computed to be 0.5 according to a preset formula. The nonlinear adjustment mode can
30 balance the global search and local search capability, the large inertia weight in the initial stage is beneficial to global exploration, and the small inertia weight in the later stage promotes local development.

 [0123] Then, the speed updating mechanism is constructed. Firstly, the Euclidean distance of the particles in the target space is computed to obtain the distance matrix. The

target space is composed of a plurality of target function values, such as completion time, and cost consumption. The distance matrix reflects the similarity between the particles. The neighborhood range of each particle is determined according to the distance matrix, for example, K nearest particles are selected as neighbors. The optimal solution is selected in the neighborhood range, and the difference between the optimal solution and the current position is computed to obtain a neighborhood guide vector. The neighborhood guide vector guides the particles to move to the neighborhood optimal solution. The neighborhood guide vector is combined with the individual optimal position item and the global optimal position item to form the complete speed update expression. The individual optimal position is the optimal position of the particle in history, and the global optimal position is the optimal position of the whole population in history. In the speed update expression, it takes into account the neighborhood guidance, individual experience and global experience completely, so that the particles can search for the optimal solution more effectively.

[0124] The particle position is updated through the updated speed, so as to obtain the candidate solution set. The difference between the function values of adjacent solutions in the candidate solution set within the target space is computed so as to obtain the congestion index. The congestion index reflects the distribution density of the solutions, and the solutions with low congestion are more beneficial to maintaining the diversity of the swarm. The non-dominated sorting is executed on the candidate solution set according to the congestion index. In the non-dominated sorting, the solutions are divided into different levels, and the lower the level is, the better the solution is. The non-dominated solution is selected to construct the external archive set. The external archive set is used for storing the non-dominated solutions, representing the currently found optimal solution set.

[0125] The newly generated non-dominated solutions are added into the external archive set and re-sorted. When the external archive set exceeds the preset capacity, the archive members will be sorted in a descending way based on the congestion index and the first 5% of solutions are reserved. The selection mechanism can effectively control the size of the archive set and reserve representative high-quality solutions.

[0126] Finally, the comprehensive assessment value is computed for the solutions in the external archive set. The comprehensive assessment value can be computed according to weights of different target functions, for example, if the completion time is more important than the cost consumption, a higher weight can be given to the completion time. The solution with the optimal comprehensive assessment value is selected as the final

solution. The resource allocation matrix and the task scheduling sequence are generated according to decision variables corresponding to the final solution, and finally the optimal resource allocation scheme and a scheduling strategy are determined. For example, the position vector of the final solution contains the resource allocation and task sorting information, and can be transformed into a resource allocation matrix and a task scheduling Gantt chart.

[0127] Through the solution of the present invention:

[0128] the solving efficiency can be improved: by nonlinearly adjusting the inertia weight and adding the neighborhood optimal solution, the local search capability of the algorithm is enhanced, and therefore, the optimal solution can be found more quickly. The solving quality can be enhanced: the individual congestion is computed based on the target spatial distance, and the selection mechanism and the maintenance strategy for the external archive set are constructed, therefore, the convergence and diversity of the algorithm are guaranteed, and a better solution can be found. The adaptability is high: this method is suitable for various multi-target resource configuration and scheduling problems, can adjust the target function and parameters according to actual requirements, and has high practical value.

[0129] In an optional embodiment, the method further includes:

[0130] based on the optimal resource configuration scheme and the scheduling strategy, executing dynamic resource adjustment by a distributed feedback controller, the distributed feedback controller implementing accurate control by a sliding mode control algorithm; constructing a multi-level anomaly detection framework to monitor an execution effect in real time, taking the feature weight matrix as a reference to identify state deviation in combination with an improved isolated forest algorithm, and analyzing a deviation transmission path based on the dynamic node association graph;

[0131] constructing a system control target based on the optimal resource configuration scheme and the scheduling strategy, transforming the system control target into a state space expression, designing a sliding mode switching function of the distributed feedback controller according to the state space expression, constructing a composite control law of an equivalent control item and a switching control item based on the sliding mode switching function, and generating a dynamic resource adjustment instruction by boundary layer smoothing processing;

[0132] executing the dynamic resource adjustment instruction and collecting real-time state data, constructing a multi-level anomaly detection framework containing the state

features, weight matrix and detection threshold, comparing the real-time state data with a reference state represented by the feature weight matrix, and computing to obtain a state deviation vector;

[0133] inputting the state deviation vector into the improved isolated forest algorithm, setting a dynamic splitting probability and an adaptive depth limit for the improved isolated forest algorithm according to the importance degree of resource features, fusing detection results of a plurality of isolated trees in a weighted voting mode, and outputting position information and degree information of an abnormal state; and

[0134] positioning an abnormal source node in the dynamic node association graph based on the position information, computing a product of the feature weight and intensity of association between adjacent nodes according to the degree information to obtain the transmission intensity of the state deviation, analyzing a diffusion path of the transmission intensity by the dynamic node association graph, and determining a key influence link.

[0135] Firstly, the current resource state information of the system is acquired, containing the CPU utilization rate, internal memory occupancy rate, network bandwidth utilization rate and the like of each node. For example, the CPU utilization rate of the node 1 is 70%, the internal memory occupancy rate is 80%, and the network bandwidth utilization rate is 50%; and the CPU utilization rate of the node 2 is 60%, the memory occupancy rate is 70%, and the network bandwidth utilization rate is 60%.

[0136] Then, the resource configuration scheme and the scheduling strategy are formulated according to the system load condition and the service quality requirement. For example, a high-priority task is scheduled to a node with sufficient resources for execution, and a low-priority task is scheduled to a node with idle resources for execution; and

[0137] Then, a system control target is constructed based on the optimal resource configuration scheme and scheduling strategy. For example, the CPU utilization rate of the node 1 is controlled to be 75% or below, the internal memory occupancy rate is controlled to be 85% or below, and the network bandwidth utilization rate is controlled to be 60% or below.

[0138] The system control target is transformed into a state spatial expression, and the sliding mode switching function of the distributed feedback controller is designed. The sliding mode switching function is used for describing the deviation between the system state and the expected state.

[0139] The composite control law of the equivalent control item and the switching control item is constructed based on the sliding mode switching function. The equivalent

control item is used for maintaining the system to be near the expected state, and the switching control item is used for overcoming the uncertainty and interference of the system. The dynamic resource adjustment instruction is generated by the boundary layer smoothing processing, for example, decreasing the CPU allocation limit of the node 1, and
 5 increasing the internal memory allocation limit of the node 2.

[0140] The dynamic resource adjustment instruction is executed, and real-time state data is collected. For example, the CPU utilization rate of the node 1 is decreased to be 72%, the internal memory occupancy rate is decreased to be 78%, and the network bandwidth utilization rate is decreased to be 55%.

10 [0141] The multi-level anomaly detection framework containing the state features, the weight matrix and the detection threshold is constructed. The state features include the CPU utilization rate, the internal memory occupancy rate, the network bandwidth utilization rate and the like. The weight matrix represents the importance of different state features. The detection threshold is used for determining whether the system state is
 15 abnormal.

[0142] The real-time state data is compared with the reference state represented by the feature weight matrix, and computation is performed to obtain the state deviation vector. For example, the CPU utilization deviation of node 1 is -3%, the memory occupancy deviation is -2%, and the network bandwidth utilization deviation is -5%.

20 [0143] The state deviation vector is inputted into the improved isolated forest algorithm. According to the importance of the resource features, the dynamic splitting probability and adaptive depth limit are set for the improved isolated forest algorithm. For example, if the importance of CPU utilization rate is higher than the internal memory occupancy rate, the splitting probability corresponding to CPU utilization rate is higher and
 25 the depth limit is smaller.

[0144] The detection results of the plurality of isolated trees are fused by the weighted voting mode, and the position information and degree information of the abnormal state are outputted. For example, the CPU utilization rate of node 1 is abnormal, and the degree is slight.

30 [0145] The abnormal source node is positioned in the dynamic node association graph based on the position information. For example, the node 1 is an abnormal source node.

[0146] The transmission intensity of state deviation is obtained by computing the product of the feature weight and intensity of association between adjacent nodes according to the degree information. For example, the intensity of transmission between the node 1

and the node 2 is 0.2.

[0147] The diffusion path of transmission intensity is analyzed by the dynamic node association graph, and thus the key influence link can be determined. For example, the abnormal state of node 1 may be transmitted to the node 2 and a node 3.

5 [0148] Through the solution of the present invention:

[0149] the resource utilization rate can be increased: by dynamically adjusting the resource allocation, the resource waste can be effectively avoided, and the overall resource utilization rate can be improved. The quality of service can be ensured: through the accurate control algorithm and multi-level anomaly detection framework, abnormal
10 situations can be found and handled in time, thereby ensuring the stability and reliability of services. The system robustness can be enhanced: by analyzing the deviation transmission path and determining the key influence link, the diffusion of abnormal states can be effectively inhibited, and the robustness and fault tolerance of the system can be enhanced.

[0150] In an optional embodiment, the method further includes:

15 [0151] optimizing a Digital Twin model based on the updated feature weight matrix and the dynamic association graph by an improved meta learning algorithm, and implementing quick iteration of the model through task decomposition and gradient accumulation; finally, analyzing a performance degradation root cause based on an improved causal inference model, and feeding back the analysis result to the hierarchical
20 reinforcement learning framework to realize dynamic optimization on the decision strategy;

[0152] initializing the feature weight matrix by the improved meta learning algorithm, each element in the feature weight matrix representing an influence weight of an input feature on an output index, acquiring a task-specific weight matrix by computing a loss function of each task, computing a meta gradient based on the task-specific weight matrix,
25 and updating a global weight matrix;

[0153] constructing the dynamic association graph based on the global weight matrix, nodes in the dynamic association graph representing system components, edges representing intensity of association among the components, screening a significant association relationship by setting a dynamic threshold, and continuously updating the
30 intensity of association by the sliding time window;

[0154] decomposing the optimization target into a plurality of subtasks and computing gradients in parallel, allocating a weight coefficient to each subtask according to the association degree of the components in the dynamic association graph, and updating parameters of the Digital Twin model by a weighted gradient accumulation mode; and

[0155] initializing a causal graph structure based on the dynamic association graph, optimizing a causal relationship by a bidirectional search algorithm and eliminating false association, quantifying the contribution degree of an influence factor to performance degradation by inverse factual inference, inputting the contribution degree as a reward
 5 signal into the hierarchical reinforcement learning framework, the hierarchical reinforcement learning framework including the macro level used for selecting macro actions and a micro level used for executing specific operations, and implementing collaborative optimization of the macro level and the micro level through the hierarchical reward mechanism.

10 [0156] Firstly, the system operation data is collected. Taking a manufacturing execution system as an example, the collected data includes device state data (e.g., running speed, temperature, and pressure), production process data (e.g., yield, qualification rate, and energy consumption) and environmental data (e.g., temperature, and humidity). If 10,000 data samples are collected, each sample contains 10 features, such as the running
 15 speed of a device A, the temperature of a device B, and the qualification rate of products.

[0157] Secondly, the feature weight matrix is initialized by the improved meta learning algorithm. Each row of the matrix represents a feature, and each column of the matrix represents an output index. Each element in the initial matrix can be set to be the same value, such as 0.1, which indicates that the influence weight of each input feature on
 20 each output index is the same. The data of a plurality of tasks are collected, such as data of different production batches or data under different operating conditions. If the data of five tasks are collected, each task contains 2,000 samples. For each task, the loss function is computed, such as a mean square error, to assess the difference between a model predicted value and a true value. The corresponding feature weight matrix is adjusted according to
 25 the loss function of each task. For example, if a certain feature contributes a lot to the loss function of a certain task, the corresponding element value of the feature in the weight matrix of the task will be increased. The meta gradient is computed based on a specific weight matrix of each task, and the global weight matrix is updated. The meta gradient indicates the updating direction and size of the global weight matrix in order that the model
 30 has high performance on all tasks.

[0158] Thirdly, the dynamic association graph is constructed based on the global weight matrix. The nodes of the dynamic association graph represent system components, such as devices, sensors and controllers. The edges represent the intensity of association between the components. The intensity of association can be determined by computing an

association coefficient between corresponding feature weight vectors of two components. The dynamic threshold is set, such as 0.8, a remarkable association relationship is screened, and the edges with intensity of association less than the threshold is removed. The sliding time window is adopted, such as a time window containing past 100 data samples, to continuously update the intensity of association, so as to reflect dynamic changes of the relationship between the system components. For example, if the running speed of the device A and the temperature of the device B are highly correlated in the past 100 samples, the intensity of association between the device A and the device B is high.

[0159] Fourthly, the optimization target is decomposed into a plurality of subtasks, and the gradients are computed in parallel. For example, a target for improving production efficiency is decomposed into subtasks for improving device utilization rate, reducing production time, improving product quality and the like. The weight coefficient is distributed to each subtask according to the association degree of components in the dynamic association graph. For example, if the intensity of association between the components related to a certain subtask is high, the subtask will be endowed with a large weight coefficient. The weighted gradient accumulation mode is adopted to update parameters of the Digital Twin model. For example, the gradient computed by each subtask is multiplied by the corresponding weight coefficient, and then the weighted gradients of all the subtasks are accumulated for updating the parameters of the Digital Twin model.

[0160] Fifthly, the causal graph structure is initialized based on the dynamic association graph. The nodes and the edges of the causal graph are the same as those of the dynamic association graph, but the edges represent causal relationships. The bidirectional search algorithm is adopted to optimize the causal relationships, and the false associations are eliminated. For example, if the running speed of the device A and the temperature of the device B have an association but have not causal relationship, the edges between them will be removed. The contributions of the influence factors to the performance degradation are quantified by inverse factual inference. For example, the influence of device A faults on production efficiency is simulated to determine the contribution of the device A faults to performance degradation. The contribution is inputted into the hierarchical reinforcement learning framework as a reward signal. The hierarchical reinforcement learning framework includes the macro level used for selecting macro actions and the micro level used for executing specific operations. For example, in the macro level, the macro action for increasing yield is selected, and in the micro level, specific operations for increasing the rotational speed of the device A are selected. The collaborative optimization of the macro

level and the micro level is achieved through the hierarchical reward mechanism. For example, if the macro action selected in the macro level and the specific operations selected in the micro level can improve production efficiency, positive rewards will be given; otherwise, negative rewards will be given.

5 [0161] Sixthly, the Digital Twin model is optimized based on the updated feature weight matrix and the dynamic association graph, and fast iteration of the model is realized by task decomposition and gradient accumulation. The above steps are repeated until the model performance reaches a preset target.

[0162] Through the solution of the present invention:

10 [0163] the model accuracy can be improved: the feature weight matrix is optimized by the meta learning algorithm, the complex relationship between the feature and the output index is captured, and thus the prediction accuracy of the Digital Twin model is improved. The construction and updating of dynamic association graph can better reflect the dynamic interaction between the system components, thereby further improving the accuracy of the model. The application of the causal inference model can identify the key factors of performance degradation, thus providing more accurate guidance for optimization decision making. The iteration of the model can be accelerated: the task decomposition and gradient accumulation strategies can parallelize the model training process, thus significantly accelerating the model iteration speed, and decreasing the model optimization cycle. The application of dynamic association graph can effectively guide the weight allocation of subtasks, thereby improving the efficiency of model training. The decision optimization can be enhanced: the hierarchical reinforcement learning framework can dynamically adjust the decision strategy according to the analysis results of causal inference model, thus realizing the collaborative optimization of the macro level and the micro level. By continuous learning and optimization, the performance of complex system is improved.

15

20

25

[0164] FIG. 2 is a flowchart of a Digital Twin based intelligent evolution system for distributed computing resources provided by an embodiment of the present invention, as shown in FIG.2, the system includes:

[0165] a first unit, configured to deploy an intelligent perception probe at each computing node of a distributed computing cluster, the intelligent perception probe acquiring computing resource operation data by a multi-task parallel acquisition algorithm; perform feature extraction on the computing resource operation data by a double-level attention mechanism, the first-level attention mechanism identifying resource state inflection points to generate an inflection feature sequence, and the second-level attention

30

mechanism extracting resource fluctuation features based on the inflection feature sequence to obtain a feature tensor; input the feature tensor into a graph neural network to construct a computing resource Digital Twin model, the graph neural network fusing a time-order convolution layer and a spatial relationship layer to establish a dynamic node association graph among the computing nodes; based on the dynamic node association graph, compute feature importance by a hybrid entropy optimization algorithm to obtain a feature weight matrix, the feature weight matrix being used for representing multi-dimensional relevancy of the resource states;

[0166] a second unit, configured to input the feature weight matrix and the dynamic node association graph into a hierarchical reinforcement learning framework, learn a global resource allocation strategy based on the feature weight matrix by an improved strategy gradient algorithm in a macro level, and optimize a local scheduling decision based on the dynamic node association graph by an improved Q learning algorithm in a micro level; and

[0167] a third unit, configured to input the global resource allocation strategy and the local scheduling decision into an attention-enhanced sequence prediction network to generate a resource utilization prediction sequence of a multi-scale time window; based on the resource utilization prediction sequence, design a dynamic fusion mechanism, and in combination with long/short-term prediction results, compute prediction weights of different time scales by an improved Bayesian optimization algorithm to obtain a fusion prediction result; and construct a multi-target optimization model according to the fusion prediction result, set a system performance index as an optimization target, and solve by a particle swarm optimization algorithm to obtain an optimal resource allocation solution and an optimal scheduling strategy.

[0168] In a third aspect,

[0169] an electronic device provided by the embodiment of the present invention includes:

[0170] a processor; and

[0171] a memory configured to store processor executable instructions;

[0172] the processor is configured to call instructions stored in the memory to execute the abovementioned method.

[0173] In a fourth aspect,

[0174] the embodiment of the present invention provides a computer-readable storage medium having computer program instructions stored therein, when executed by a processor, the computer program instructions execute the abovementioned method.

[0175] The present invention can be a method, an apparatus, a system, and/or a computer program product. The computer program product can include a computer-readable storage medium that contains computer-readable program instructions for executing the present invention in various aspects.

5 [0176] Finally, it is to be noted that: the above embodiments are only used for illustrating the technical solution of the present invention, not to limit it. Although the present invention is described in detail with reference to the aforementioned embodiments, those of ordinary skill in the art should understand that it may still modify the technical solutions recorded in the aforementioned embodiments, or replace some or all of the
10 technical features thereof. These modifications or substitutions do not make the essence of the corresponding technical solution out of the scope of the technical solutions of each embodiment of the present invention.

CLAIMS

We claim:

1. A Digital Twin based intelligent evolution method for distributed computing resources, comprising:

5 deploying an intelligent perception probe at each computing node of a distributed computing cluster, the intelligent perception probe acquiring computing resource operation data by a multi-task parallel acquisition algorithm; performing feature extraction on the computing resource operation data by a double-level attention mechanism, the first-level attention mechanism identifying resource state inflection points to generate an inflection
10 feature sequence, and the second-level attention mechanism extracting resource fluctuation features based on the inflection feature sequence to obtain a feature tensor; inputting the feature tensor into a graph neural network to construct a computing resource Digital Twin model, the graph neural network fusing a time-order convolution layer and a spatial relationship layer to establish a dynamic node association graph among the computing
15 nodes; based on the dynamic node association graph, computing feature importance by a hybrid entropy optimization algorithm to obtain a feature weight matrix, the feature weight matrix being used for representing multi-dimensional relevancy of the resource states;

 inputting the feature weight matrix and the dynamic node association graph into a hierarchical reinforcement learning framework, learning a global resource allocation
20 strategy based on the feature weight matrix by an improved strategy gradient algorithm in a macro level, and optimizing a local scheduling decision based on the dynamic node association graph by an improved Q learning algorithm in a micro level; and

 inputting the global resource allocation strategy and the local scheduling decision into an attention-enhanced sequence prediction network to generate a resource utilization
25 prediction sequence of a multi-scale time window; based on the resource utilization prediction sequence, designing a dynamic fusion mechanism, and in combination with long/short-term prediction results, computing prediction weights of different time scales by an improved Bayesian optimization algorithm to obtain a fusion prediction result; and constructing a multi-target optimization model according to the fusion prediction result,
30 setting a system performance index as an optimization target, and solving by a particle swarm optimization algorithm to obtain an optimal resource allocation solution and an optimal scheduling strategy.

2. The method according to claim 1, wherein the inputting the feature tensor into a graph neural network to construct a computing resource Digital Twin model, the graph

neural network fusing a time-order convolution layer and a spatial relationship layer to establish a dynamic node association graph among the computing nodes; based on the dynamic node association graph, computing feature importance by a hybrid entropy optimization algorithm to obtain a feature weight matrix, the feature weight matrix being
 5 used for representing multi-dimensional relevancy of the resource states comprises:

inputting the feature tensor into the graph neural network that fuses the time-order convolution layer and the spatial relationship layer, the time-order convolution layer being of a multi-scale parallel convolution structure, performing causal filling to keep the sequence length unchanged so as to obtain multi-scale time-order features; the spatial
 10 relationship layer adopting a dynamic attention mechanism, computing an inter-node attention weight based on a learnable parameter matrix, and performing adaptive aggregation on the node features according to the inter-node attention weight;

constructing the dynamic node association graph among the computing nodes based on the output of the graph neural network, computing a Pearson association coefficient matrix and a resource calling dependency matrix between the nodes by a sliding time
 15 window mechanism, fusing the Pearson association coefficient matrix with the resource calling dependency matrix by the adaptive weight coefficient to obtain a comprehensive intensity of association matrix, and dynamically updating the comprehensive intensity of association matrix based on an exponential attenuation mode so as to obtain the dynamic
 20 node association graph;

based on the dynamic node association graph, computing the feature importance by the hybrid entropy optimization algorithm, the hybrid entropy optimization algorithm computing an information entropy of a single feature through kernel density estimation, then computing a cross entropy between the feature and a target variable, computing
 25 conditional mutual information based on the information entropy and the cross entropy, and normalizing the conditional mutual information to obtain the feature weight matrix;

dynamically optimizing the feature weight matrix, adding a time attenuation factor to adjust the weight according to feature timeliness, computing a weight gradient based on prediction error loss, and updating the weight by a gradient descent method of a momentum
 30 item, the momentum item being used for smoothing the weight updating process; and

feeding back the updated feature weight matrix to the graph neural network for guiding dynamic weighted aggregation of the node features in the spatial relationship layer, thereby realizing iterative optimization of the computing resource Digital Twin model, the feature weight matrix representing the multi-dimensional relevancy of the resource states.

3. The method according to claim 1, wherein the inputting the feature weight matrix and the dynamic node association graph into a hierarchical reinforcement learning framework, learning a global resource allocation strategy based on the feature weight matrix by an improved strategy gradient algorithm in a macro level, and optimizing a local scheduling decision based on the dynamic node association graph by an improved Q learning algorithm in a micro level comprises:

inputting the feature weight matrix and the dynamic node association graph into the hierarchical reinforcement learning framework, the hierarchical reinforcement learning framework comprising the macro level and the micro level, the hierarchical reinforcement learning framework implementing strategy collaboration in the macro level and the micro level based on a hierarchical reward mechanism, and the hierarchical reward mechanism constructing a strategy learning target by a weighted combination of a global reward value and a local reward value;

learning the global resource allocation strategy based on the feature weight matrix by the improved strategy gradient algorithm in the macro level, the improved strategy gradient algorithm mapping feature weight information into resource allocation action probability by a strategy network, adding a trust domain constraint to limit strategy update step length, constructing an optimization target function by a new and old strategy ratio cutting mode, and updating strategy network parameters in combination with a time-attenuated adaptive learning rate;

optimizing the local scheduling decision based on the dynamic node association graph by the improved Q learning algorithm in the micro level, the improved Q learning algorithm constructing a dual Q network structure, comprising a principal value network and an auxiliary value network, extracting state features of a target node and neighbor nodes based on the dynamic node association graph, combining outputs of the two value networks by a learnable fusion coefficient, and performing sample learning by a time-order difference error based priority experience playback mechanism;

implementing the collaborative decision-making in the macro level and the micro level by a bidirectional information flow mechanism, transforming the global resource allocation strategy learned in the macro level into local decision constraints for guiding action selection in the micro level, and feeding back the local state information in the micro level to the macro level after pooling operation aggregation; and

constructing a cross-level value assessment function, performing weighted combination on the global value assessment in the macro level and the local value

assessment in the micro level, and performing joint optimization on the global resource allocation strategy and the local scheduling decision based on the cross-level value assessment function.

4. The method according to claim 1, wherein the based on the resource utilization
 5 prediction sequence, designing a dynamic fusion mechanism, and in combination with
 long/short-term prediction results, computing prediction weights of different time scales by
 an improved Bayesian optimization algorithm to obtain a fusion prediction result; and
 constructing a multi-target optimization model according to the fusion prediction result,
 setting a system performance index as an optimization target, and solving by a particle
 10 swarm optimization algorithm to obtain an optimal resource allocation solution and an
 optimal scheduling strategy comprises:

dividing the resource utilization prediction sequence into a short-term prediction
 sequence and a long-term prediction sequence according to different time spans, the
 short-term prediction sequence containing prediction values of continuous time points, and
 15 the long-term prediction sequence containing prediction values of interval sampling;

inputting the short-term prediction sequence and the long-term prediction sequence
 into the improved Bayesian optimization algorithm, the improved Bayesian optimization
 algorithm constructing a dynamic Gaussian process kernel function containing a
 time-varying noise item, adaptively adjusting exploration parameters based on prediction
 20 errors, and updating Gaussian process model parameters online by the sliding time window,
 so as to obtain fusion weights of the short-term prediction sequence and the long-term
 prediction sequence;

performing weighted combination on the fusion weights with the short-term prediction
 sequence and the long-term prediction sequence to obtain a fusion prediction result,
 25 constructing a multi-target optimization model based on the fusion prediction result, the
 multi-target optimization model comprising a system throughput target, a resource
 utilization rate target and an energy consumption efficiency target, and setting a resource
 capacity upper limit constraint, a task completion deadline constraint and a load balance
 degree constraint; and

30 utilizing the particle swarm algorithm to solve the multi-target optimization model, the
 particle swarm algorithm adaptively adjusting inertia weights through nonlinear regulation
 factors, adding a neighborhood optimal solution into a speed updating formula to enhance
 local search capability, computing individual congestions based on target spatial distances
 and constructing a selection mechanism, storing non-dominated solution through an

external archive set, and generating the optimal resource allocation scheme and the scheduling strategy based on the solving result of the particle swarm algorithm.

5 5. The method according to claim 4, wherein the utilizing the particle swarm algorithm to solve the multi-target optimization model, the particle swarm algorithm adaptively adjusting inertia weights through nonlinear regulation factors, adding a neighborhood optimal solution into a speed updating formula to enhance local search capability, computing individual congestions based on target spatial distances and constructing a selection mechanism, storing non-dominated solution through an external archive set, and generating the optimal resource allocation scheme and the scheduling
10 strategy based on the solving result of the particle swarm algorithm comprises:

initializing a position vector and a speed vector of a particle swarm, designing the nonlinear regulation factors, the nonlinear regulation factors being of a composite form of a cosine function and a power function, and computing a current inertia weight according to the ratio of the current number of iterations to the maximum number of iterations;

15 constructing a speed updating mechanism based on the current inertia weight, computing the Euclidean distance of particles in a target space to obtain a distance matrix, determining the neighborhood range of each particle according to the distance matrix, selecting an optimal solution in the neighborhood range, computing the difference between the optimal solution and the current position to obtain a neighborhood guide vector, and
20 combining the neighborhood guide vector with an individual optimal position item and a global optimal position item to obtain a complete speed updating expression;

updating the particle position by the speed updating expression to obtain a candidate solution set, computing the function value difference of adjacent solutions in the candidate solution set in the target space to obtain a congestion index, performing non-dominated
25 sorting on the candidate solution set according to the congestion index, and selecting a non-dominated solution to construct the external archive set;

adding the newly generated non-dominated solution into the external archive set and re-sorting, when the external archive set exceeds a preset capacity, sorting archive members in a descending way based on the congestion index and reserving the first 5% of solutions,
30 and storing high-quality solutions with convergence and diversity through the external archive set; and

computing a comprehensive assessment value for the solutions in the external archive set, selecting the solution with the optimal comprehensive assessment value as a final solution, generating a resource allocation matrix and a task scheduling sequence according

to a decision variable corresponding to the final solution, and determining the optimal resource allocation scheme and the scheduling strategy based on the resource allocation matrix and the task scheduling sequence.

6. The method according to claim 1, further comprising:

5 based on the optimal resource configuration scheme and the scheduling strategy, executing dynamic resource adjustment by a distributed feedback controller, the distributed feedback controller implementing accurate control by a sliding mode control algorithm; constructing a multi-level anomaly detection framework to monitor an execution effect in real time, taking the feature weight matrix as a reference to identify state deviation in
10 combination with an improved isolated forest algorithm, and analyzing a deviation transmission path based on the dynamic node association graph;

 constructing a system control target based on the optimal resource configuration scheme and the scheduling strategy, transforming the system control target into a state space expression, designing a sliding mode switching function of the distributed feedback
15 controller according to the state space expression, constructing a composite control law of an equivalent control item and a switching control item based on the sliding mode switching function, and generating a dynamic resource adjustment instruction by boundary layer smoothing processing;

 executing the dynamic resource adjustment instruction and collecting real-time state
20 data, constructing a multi-level anomaly detection framework containing the state features, weight matrix and detection threshold, comparing the real-time state data with a reference state represented by the feature weight matrix, and computing to obtain a state deviation vector;

 inputting the state deviation vector into the improved isolated forest algorithm, setting
25 a dynamic splitting probability and an adaptive depth limit for the improved isolated forest algorithm according to the importance degree of resource features, fusing detection results of a plurality of isolated trees in a weighted voting mode, and outputting position information and degree information of an abnormal state; and

 positioning an abnormal source node in the dynamic node association graph based on
30 the position information, computing a product of the feature weight and intensity of association between adjacent nodes according to the degree information to obtain the transmission intensity of the state deviation, analyzing a diffusion path of the transmission intensity by the dynamic node association graph, and determining a key influence link.

7. The method according to claim 1, further comprising:

optimizing a Digital Twin model based on the updated feature weight matrix and the dynamic association graph by an improved meta learning algorithm, and implementing quick iteration of the model through task decomposition and gradient accumulation; finally, analyzing a performance degradation root cause based on an improved causal inference
 5 model, and feeding back the analysis result to the hierarchical reinforcement learning framework to realize dynamic optimization on the decision strategy;

initializing the feature weight matrix by the improved meta learning algorithm, each element in the feature weight matrix representing an influence weight of an input feature on an output index, acquiring a task-specific weight matrix by computing a loss function of
 10 each task, computing a meta gradient based on the task-specific weight matrix, and updating a global weight matrix;

constructing the dynamic association graph based on the global weight matrix, nodes in the dynamic association graph representing system components, edges representing intensity of association among the components, screening a significant association
 15 relationship by setting a dynamic threshold, and continuously updating the intensity of association by the sliding time window;

decomposing the optimization target into a plurality of subtasks and computing gradients in parallel, allocating a weight coefficient to each subtask according to the association degree of the components in the dynamic association graph, and updating
 20 parameters of the Digital Twin model by a weighted gradient accumulation mode; and

initializing a causal graph structure based on the dynamic association graph, optimizing a causal relationship by a bidirectional search algorithm and eliminating false association, quantifying the contribution degree of an influence factor to performance degradation by inverse factual inference, inputting the contribution degree as a reward
 25 signal into the hierarchical reinforcement learning framework, the hierarchical reinforcement learning framework comprising the macro level used for selecting macro actions and a micro level used for executing specific operations, and implementing collaborative optimization of the macro level and the micro level through the hierarchical reward mechanism.

30 8. A Digital Twin based intelligent evolution system for distributed computing resources, being applied to the method according to any one of claims 1 to 7, comprising:

a first unit, configured to deploy an intelligent perception probe at each computing node of a distributed computing cluster, the intelligent perception probe acquiring computing resource operation data by a multi-task parallel acquisition algorithm; perform

feature extraction on the computing resource operation data by a double-level attention mechanism, the first-level attention mechanism identifying resource state inflection points to generate an inflection feature sequence, and the second-level attention mechanism extracting resource fluctuation features based on the inflection feature sequence to obtain a feature tensor; input the feature tensor into a graph neural network to construct a computing resource Digital Twin model, the graph neural network fusing a time-order convolution layer and a spatial relationship layer to establish a dynamic node association graph among the computing nodes; based on the dynamic node association graph, compute feature importance by a hybrid entropy optimization algorithm to obtain a feature weight matrix, the feature weight matrix being used for representing multi-dimensional relevancy of the resource states;

a second unit, configured to input the feature weight matrix and the dynamic node association graph into a hierarchical reinforcement learning framework, learn a global resource allocation strategy based on the feature weight matrix by an improved strategy gradient algorithm in a macro level, and optimize a local scheduling decision based on the dynamic node association graph by an improved Q learning algorithm in a micro level; and

a third unit, configured to input the global resource allocation strategy and the local scheduling decision into an attention-enhanced sequence prediction network to generate a resource utilization prediction sequence of a multi-scale time window; based on the resource utilization prediction sequence, design a dynamic fusion mechanism, and in combination with long/short-term prediction results, compute prediction weights of different time scales by an improved Bayesian optimization algorithm to obtain a fusion prediction result; and construct a multi-target optimization model according to the fusion prediction result, set a system performance index as an optimization target, and solve by a particle swarm optimization algorithm to obtain an optimal resource allocation solution and an optimal scheduling strategy.

9. An electronic device, comprising a processor; and

a memory configured to store processor executable instructions; and

the processor is configured to call instructions stored in the memory to execute the method according to any one of claims 1 to 7.

10. A computer-readable storage medium having computer program instructions stored therein, wherein when executed by a processor, the computer program instructions execute the method according to any one of claims 1 to 7.