United States Patent Application Publication	20250258486
Kind Code	A1
Publication Date	August 14, 2025
Inventor(s)	PAN; Fucheng et al.

TASK-LEVEL COOPERATIVE OPTIMIZATION DISPATCHING METHOD SUPPORTING MULTI-CLOUD DISCONNECTION DISASTER RECOVERY

Abstract

A task-level cooperative optimization dispatching method supporting multi-cloud disconnection disaster recovery is provided. Since cooperative dispatching is required for different resource requirements in the abnormal stage of a network, a cooperative computing system is formed by nodes of a production shop, production tasks are transmitted and processed between the nodes according to technological requirements, and task-level cooperative optimization dispatching is conducted through a CoopDispatch method in case of disconnection to ensure continuous production. The CoopDispatch method has the characteristic of high dependence on specific resources for manufacturing tasks. Data transmission, storage and calculation are achieved through a LocalAny mechanism, which can reduce task delay, improve the cooperation efficiency among the tasks and satisfy the needs of manufacturing enterprises for disconnection disaster recovery in the multi-cloud mode.

Inventors:	**PAN; Fucheng (Shenyang, Liaoning, CN), SHI; Haibo (Shenyang, Liaoning, CN), HU; Guoliang (Shenyang, Liaoning, CN), LI; Xin (Shenyang, Liaoning, CN), LI; Peng (Shenyang, Liaoning, CN)**
Applicant:	**SHENYANG INSTITUTE OF AUTOMATION, CHINESE ACADEMY OF SCIENCES** (Shenyang, Liaoning, CN)

Family ID:	**90043289**
Appl. No.:	**18/858777**
Filed (or PCT Filed):	**January 16, 2024**
PCT No.:	**PCT/CN2024/072423**

Foreign Application Priority Data

CN 202311756761.5 Dec. 20, 2023

Publication Classification

Int. Cl.: **G05B19/418** (20060101)

U.S. Cl.:

CPC **G05B19/41885** (20130101); **G05B19/4185** (20130101);

Background/Summary

TECHNICAL FIELD

[0001] The present invention relates to the technical field of computer system information, in particular to a task-level cooperative optimization dispatching method supporting multi-cloud disconnection disaster recovery.

BACKGROUND

[0002] The rapid development of a new generation of information technology provides good technical support for the technology integration innovation of the manufacturing industry, and focuses on promoting the high-quality development of the manufacturing industry, strengthening the industrial foundation and technology innovation capacity, promoting the integrated development of the advanced manufacturing industry and modern service industry, accelerating the construction of a manufacturing power, creating an industrial Internet platform and expanding "intelligence plus" to enable the transformation and upgrading of the manufacturing industry to become the main development direction of the new round of industrial revolution. Under this background, the intelligent decision of enterprises needs a new application innovation carrier. With the transformation of the manufacturing industry and the integration of digital economy wave, the integration innovation of information technologies such as cloud computing, the Internet of things and big data with the manufacturing technology and industrial knowledge has continued to intensify, and an industrial Internet platform has emerged. Business cloudification of the enterprises has become a trend.

[0003] In the traditional manufacturing mode, the business system of the enterprise is operated locally, and may not be affected by the fault of a peripheral network. However, after cloudification of the business system of the enterprise, if the previous "resource-centered" mode of requesting at a client and response at a server is still used, the business system of the enterprise will be affected once a network of an enterprise end and a cloud fails. On the basis of the prior art, the present invention focuses on solving the problem of how to ensure the cooperative optimization of tasks due to the highly dependent characteristics of specific resources of a manufacturing task when a local area network of a production workshop is normal and a network among multiple clouds is disconnected under a multi-cloud mode, and fills the technical gap in this field.

SUMMARY

[0004] In view of the defects of the prior art, the present invention provides a task-level cooperative optimization dispatching method supporting multi-cloud disconnection disaster recovery. Considering that the "resource-centered" real-time request-response mode of the traditional production management and control system is not applicable to a multi-cloud

environment and a large number of frequent data interactions are needed between manufacturing nodes in the production management and control system in the multi-cloud environment, the method of the present invention provides task-level cooperative optimization dispatching through a CoopDispatch method in case of disconnection to ensure continuous production, and achieves data transmission, storage and calculation through a LocalAny mechanism, which can reduce task delay, improve the cooperation efficiency among tasks and satisfy the needs of manufacturing enterprises for disconnection disaster recovery in the multi-cloud mode.

[0005] To realize the above purpose, the present invention adopts the technical solution: a task-level cooperative optimization dispatching method supporting multi-cloud disconnection disaster recovery comprises the following steps: [0006] assigning, by a manufacturing node, a task to other matched manufacturing nodes through an end-to-end communication channel not dependent on a service; [0007] conducting cooperative optimization dispatching for the task between the manufacturing node and other manufacturing nodes by a CoopDispatch method; [0008] synchronizing task processing results of the manufacturing node and other manufacturing nodes among the manufacturing nodes by a LocalAny mechanism.

[0009] The manufacturing node is a client of an edge side for workshops of manufacturing enterprises to carry a production management and control system in a multi-cloud environment.

[0010] The multi-cloud comprises a cloud formed after cloudification of an enterprise management and control business and a sub-cloud of a production workshop of each manufacturing enterprise.

[0011] The CoopDispatch method calculates a resource-limited minimum task execution average time in a cooperative computing system composed of a plurality of manufacturing nodes to cooperatively dispatch the task to improve the overall revenue of the cooperative computing system, comprising the following steps: [0012] 1) dividing tasks in the cooperative computing system CS into class |M|; representing a task set as Task; for the tasks in class m, m∈|M|, and processing the tasks only in the task set Task.sub.m∈Task where resource R.sub.i exists; for the current manufacturing node, an arrival rate A.sub.i.sup.m(t) of a request for the tasks in class m to other manufacturing nodes at time t satisfies:

[00001] $0 \le A_i^m(t) \le A_m^{\max} * {}_{\{T_i \in \text{Task}_m\}}$ [0013] wherein A.sub.m.sup.max represents a maximum arrival rate of the tasks in class m, i represents an ith task of the tasks in class τ.sub. {T.sub.i.sub.∈Task.sub.m.sub.} is a flag bit for representing whether the task T.sub.i that can be processed by the resource R.sub.i is in the task set Task.sub.m, i.e.:

[00002] ${}_{\{T_i \in \text{Task}_m\}} = \begin{cases} 1, & \forall_i \in \text{Task}_m \\ 0, & \forall_i \text{ .Math. } \text{Task}_m \end{cases}$ [0014] the manufacturing node processes multiple tasks at the same time, the tasks in class m at time t are stored in a queue Q.sub.i.sup.m(t), and all task queues {Q.sub.i.sup.m(t), r.sub.i∈R.sub.i, Task.sub.m∈Task} constitute the following queue matrix Q(t):

[00003] $Q(t) = \begin{bmatrix} Q_1^1(t) & Q_1^2(t) & \text{.Math.} & Q_1^{\text{.Math. } M \text{ .Math.}}(t) \\ Q_2^1(t) & Q_2^2(t) & \text{.Math.} & Q_2^{\text{.Math. } M \text{ .Math.}}(t) \\ \text{.Math.} & \text{.Math.} & \text{.Math.} & \text{.Math.} \\ Q_I^1(t) & Q_I^2(t) & \text{.Math.} & Q_I^{\text{.Math. } M \text{ .Math.}}(t) \end{bmatrix}$ [0015] r.sub.i represents a resource in resource set R.sub.i, and Q.sub.I.sup.|M|(t) represents a last task of the processed tasks in class m with a queue waiting number of |M|; Q.sub.i.sup.m(t) represents a task in class m processed by the resource r.sub.i, i=1 . . . . I; [0016] the number of requests of the task in class m received at time t is defined as a.sub.i.sup.m(t), which satisfies:

[00004] $0 \le a_i^m(t) \le A_i^m(t) * \quad t$ [0017] wherein Δt represents a time interval, which is a constant; [0018] the task execution average time is as follows:

[00005] $a_i^m = \lim_{t \text{ .fwdarw. } \infty} \frac{1}{t} \text{ .Math. } \sum_{t=0}^{t=1} E\{a_i^m(\ _{\{T_i \in \text{Task}_m\}})\}$ [0019] wherein α is an empirical constant, E{ } represents an expectation, which indicates the expectation of the task execution average time a.sub.i.sup.m; [0020] 2) assigning, by the cooperative computing system CS, resources required by the tasks to process the tasks, and assigning the resource CS.sub.i corresponding to the ith task to the resource requested by the task in class m, r.sub.i.sup.m(t): [00006] $\text{.Math. } \sum_m r_i^m(t) \le R_i$ [0021] updating a task processing queue by the manufacturing node through the following model:

[00007] $Q_i^m(t+1) = \max(Q_i^m(t) - r_i^m(t) * \ _m)$ [0022] wherein ϑ.sub.m represents the number of the tasks in class m that can be processed by unit resource in a set cycle; [0023] 3) constructing a comprehensive benefit model according to the task execution average time a.sub.i.sup.m and the resource r.sub.i.sup.m(t), and obtaining a combination of minimum execution average time a.sub.i.sup.m and resource r.sub.i.sup.m(t) by comprehensive benefit maximization of the collaborative computing system CS to achieve cooperative optimization dispatching.

[0024] In step 3), achieving cooperative optimization dispatching by comprehensive benefit maximization of the collaborative computing system CS comprises the following steps: [0025] (1) when the task of the manufacturing node is received by other manufacturing nodes, the manufacturing node obtains the following benefits:

[00008] $\int 1^{(a_i^m)} = \log(1 + \ _m * a_i^m)$ [0026] wherein § 1.sup.(a.sup.i.sup.m.sup.) represents the first part of benefit, β.sub.m is a request revenue constant of the task in class m and a.sub.i.sup.m is the task execution average time; [0027] (2) the number n.sub.m(t) of requests of the task in class m processed by the manufacturing node at time t is:

[00009] $n_m(t) = \text{ .Math. } \sum_i r_i^m(t) * \ _m$ [0028] ϑ.sub.m represents the number of the tasks in class m that can be processed by unit resource in a set cycle; [0029] corresponding revenue is:

[00010] $\int 2^{(n_m)} = \ _m * n_m$ [0030] wherein n.sub.m is a time expectation of n.sub.m(t), and θ.sub.m is a corresponding revenue constant; § 2.sup.(n.sup.m.sup.) represents the second part of benefit, which depends on the number of served requests; [0031] (3) the comprehensive benefit model is:

[00011] $B = \text{ .Math. } \sum_{i,m} \int 1^{(a_i^m)} + \text{ .Math. } \sum_m \int 2^{(n_m)}$ [0032] wherein i represents the ith task of the tasks in class m; [0033] (4) to maximize the comprehensive benefit to achieve cooperative optimization, a.sub.i.sup.m obtained is the minimum execution average time when B reaches a maximum value.

[0034] Synchronizing task processing results of the manufacturing node and other manufacturing nodes among the manufacturing nodes by a LocalAny mechanism comprises the following steps: [0035] when the manufacturing node receives the tasks of other nodes, with the first part of benefit § 1.sup.(a.sup.i.sup.m.sup.) being nonlinear, adding a transform constant variable y.sub.i.sup.m(t) and a constant variable transposition queue {Y.sub.i.sup.m(t), ∀i, m} to convert task optimization dispatching into a linear problem:

[00012] $(\ \ (t)) = \text{ .Math. } \sum_{i,m} E\{Q_i^m(t) * w_m * a_i^m(t) - Y_i^m(t) * \partial_m * a_i^m(t) \text{ .Math. } \ \ (t)\}$ [0036] wherein w.sub.m and ϑ.sub.m are constant variables, Φ(Ω(t)) represents a maximum comprehensive benefit value, Ω(t) represents a difference between a current comprehensive benefit and an average comprehensive benefit, E{ } represents an expectation which here represents an expected value of benefit, and a.sub.i.sup.m is the task execution average time; [0037] substituting the combination of the execution average time a.sub.i.sup.m and the resource r.sub.i.sup.m(t) obtained by comprehensive benefit maximization of the cooperative computing system CS in step 3) into Φ(Ω(t)), and minimizing Φ(Ω(t)) to obtain an optimized task dispatching strategy;

[00013] $a_i^m(t) = \begin{cases} A_i^m(t), & Q_i^m(t) \le Y_i^m(t) \\ 0, & \text{other} \end{cases}$ [0038] when a real queue Q.sub.i.sup.m(t) is not

longer than the constant variable transposition queue, receiving, by the manufacturing node, all requests of the tasks in class m that reach at time t, otherwise waiting.
[0039] The present invention has the following beneficial effects and advantages: [0040] 1. The task-level cooperative optimization dispatching through the CoopDispatch method in case of disconnection in the method of the present invention can ensure continuous production. [0041] 2. The LocalAny mechanism is adopted by the present invention to achieve data transmission, storage and calculation, which can reduce task delay, improve the cooperation efficiency among tasks and satisfy the needs of manufacturing enterprises for disconnection disaster recovery in the multi-cloud mode.

Description

DESCRIPTION OF DRAWINGS
[0042] FIG. **1** is a schematic diagram of a method of the present invention.
DETAILED DESCRIPTION
[0043] The present invention will be further described in detail below in combination with the drawings and the embodiments.
[0044] FIG. **1** is a schematic diagram of a method of the present invention.
[0045] The present invention relates to a task-level cooperative optimization dispatching method supporting multi-cloud disconnection disaster recovery. With the continuous development and popularization of the cloud computing technology, cloudification of the manufacturing enterprises has become a trend. After enterprise cloudification, when a local area network of a production workshop is normal and a network of the multi-cloud is disconnected, how to ensure normal and sustainable production has become a technical problem concerned in the industry. In view of the problem that cooperative dispatching is required for different resource requirements in the abnormal stage of a network, a cooperative computing system is formed by nodes of a production shop, production tasks are transmitted and processed between the nodes according to technological requirements, and task-level cooperative optimization dispatching is conducted through a CoopDispatch method in case of disconnection to ensure continuous production. The CoopDispatch method adopted by the present invention has the characteristic of high dependence on specific resources for manufacturing tasks. Data transmission, storage and calculation are achieved through a LocalAny mechanism, which can reduce task delay, improve the cooperation efficiency among the tasks and satisfy the needs of manufacturing enterprises for disconnection disaster recovery in the multi-cloud mode.
[0046] A task-level cooperative optimization dispatching method supporting multi-cloud disconnection disaster recovery comprises the following steps: [0047] assigning, by a manufacturing node, a computing-intensive and resource-sensitive task to matched manufacturing nodes through an end-to-end communication channel not dependent on a service; [0048] wherein computing-intensive means that computing resources required for processing the task are high, such as special resources of GPU resources and high computing power; [0049] resource-sensitive means that processing of the task requires specific manufacturing resources, such as equipment with special production capacity or production lines with specific technologies according to the production technologies; [0050] conducting cooperative optimization dispatching for the task among the manufacturing nodes by a CoopDispatch method; [0051] synchronizing task processing results among the manufacturing nodes by a LocalAny mechanism.
[0052] The manufacturing node is a client of an edge side for workshops of manufacturing enterprises to carry a production management and control system in a multi-cloud environment.
[0053] The multi-cloud comprises a cloud formed after cloudification of an enterprise management and control business and a sub-cloud of a production workshop of each manufacturing enterprise.

[0054] The CoopDispatch method calculates a resource-limited minimum task execution average time in a cooperative computing system to cooperatively dispatch the task to improve the overall revenue of the system, comprising the following steps: [0055] dividing tasks in the cooperative computing system CS into class |M|; representing a task set as Task; for the tasks in class m, m∈|M|, and processing the tasks only in the task set Task.sub.m∈Task where resource R.sub.i exists; for the current manufacturing node, an arrival rate A.sub.i.sup.m(t) of a request for the tasks in class m to other manufacturing nodes at time t satisfies:

[00014] $0 \le A_i^m(t) \le A_m^{max} * {}_{\{T_i \in \mathrm{Task}_m\}}$ [0056] wherein A.sub.m.sup.max represents a maximum arrival rate of the tasks in class m, i represents an ith task of the tasks in class m, τ.sub.{T.sub.i.sub.∈Task.sub.m.sub.} is a flag bit for representing whether the task T.sub.i that can be processed by the resource R.sub.i is in the task set Task.sub.m, i.e.:

[00015] ${}_{\{T_i \in \mathrm{Task}_m\}} = \{ \begin{matrix} 1, \forall_i \in \mathrm{Task}_m \\ 0, \forall_i \text{ .Math. } \mathrm{Task}_m \end{matrix}$ [0057] the manufacturing node processes multiple tasks at the same time, the tasks in class m at time t are stored in a queue Q.sub.i.sup.m(t), and all task queues {Q.sub.i.sup.m(t), r.sub.i∈ R.sub.i, Task.sub.m∈Task} constitute the following queue matrix Q(t):

[00016] $Q(t) = \begin{bmatrix} Q_1^1(t) & Q_1^2(t) & \text{.Math.} & Q_1^{\text{.Math. } M \text{ .Math.}}(t) \\ Q_2^1(t) & Q_2^2(t) & \text{.Math.} & Q_2^{\text{.Math. } M \text{ .Math.}}(t) \\ \text{.Math.} & \text{.Math.} & \text{.Math.} & \text{.Math.} \\ Q_I^1(t) & Q_I^2(t) & \text{.Math.} & Q_I^{\text{.Math. } M \text{ .Math.}}(t) \end{bmatrix}$ [0058] r.sub.i represents a resource in resource set R.sub.i, and Q.sub.I.sup.|M|(t) represents a last task of the processed tasks in class m with a queue waiting number of |M|; Q.sub.i.sup.m(t) represents a queue of tasks in class m processed by the resource r.sub.i, i=1 . . . . I; [0059] the number of requests of the task in class m received at time t is defined as a.sub.i.sup.m(t), which satisfies:

[00017] $0 \le a_i^m(t) \le A_i^m(t) * \quad t$ [0060] wherein Δt represents a time interval, which is a constant; [0061] the task execution average time is as follows:

[00018] $a_i^m = \lim_{t \text{ .fwdarw. } \infty} \frac{1}{t} \text{ .Math.} \sum_{t=0}^{t=1} E\{a_i^m({}_{\{T_i \in \mathrm{Task}_m\}})\}$ [0062] wherein α is an empirical constant, and E represents an expectation of the task execution average time a.sub.i.sup.m; [0063] assigning, by the cooperative computing system CS, resources required by the tasks to process the tasks, and assigning the resource CS.sub.i corresponding to the ith task to the resource requested by the task in class m, r.sub.i.sup.m(t):

[00019] $\text{.Math.}_m r_i^m(t) \le R_i$ [0064] updating a task processing queue by the manufacturing node through the following model:

[00020] $Q_i^m(t+1) = \max(Q_i^m(t) - r_i^m(t) * {}_m)$ [0065] wherein ϑ.sub.m represents the number of the tasks in class m that can be processed by unit resource in a certain cycle; [0066] constructing a comprehensive benefit model according to the task execution average time a.sub.i.sup.m and the resource r.sub.i.sup.m(t), and obtaining a combination of minimum execution average time a.sub.i.sup.m and resource r.sub.i.sup.m(t) by comprehensive benefit maximization of the collaborative computing system CS to achieve cooperative optimization dispatching.

[0067] Finally, the comprehensive benefit maximization of the collaborative computing system CS is computed to achieve the purpose of cooperative optimization. The benefits here can be divided into two parts:

[0068] When the task of the manufacturing node is received by other manufacturing nodes, the manufacturing node obtains the following benefits:

[00021] $\int 1^{(a_i^m)} = \log(1 + {}_m * a_i^m)$ [0069] wherein § 1.sup.(a.sup.i.sup.m.sup.) represents the first part of benefit, β.sub.m is a request revenue constant of the task in class m and a.sub.i.sup.m is the

task execution average time; [0070] the number n.sub.m(t) of requests of the task in class m processed by the manufacturing node at time t is:

[00022] $n_m(t) =$ .Math. $r_i^m(t) *$ $_m$ [0071] corresponding revenue is:

[00023] $\int 2^{(n_m)} =$ $_m * n_m$ [0072] wherein n.sub.m is a time expectation of n.sub.m(t), and θ.sub.m is a corresponding revenue constant; § 2.sup.(n.sup.m.sup.) represents the second part of benefit, which depends on the number of served requests; [0073] the comprehensive benefit model is:

[00024] $B =$ .Math. $\int 1^{(a_i^m)} +$ .Math. $\int 2^{(n_m)}$ [0074] wherein i represents the ith task of the tasks in class m; [0075] to maximize the comprehensive benefit to achieve cooperative optimization, a.sub.i.sup.m obtained is the minimum execution average time when B reaches a maximum value. [0076] When the manufacturing node receives the tasks of other nodes, with the first part of benefit § 1.sup.(a.sup.i.sup.m.sup.) being nonlinear, adding a transform constant variable y.sub.i.sup.m(t) and a constant variable transposition queue {Y.sub.i.sup.m(t), ∀i, m} to convert task optimization dispatching into a linear problem:

[00025] ((t)) = .Math. $E\{Q_i^m(t) * w_m * a_i^m(t) - Y_i^m(t) * \partial_m * a_i^m(t) \mid$ (t)} [0077] wherein w.sub.m and ϑ.sub.m are constant variables, Φ(Ω(t)) represents a maximum comprehensive benefit value, Ω(t) represents a difference between a current comprehensive benefit and an average comprehensive benefit, E represents an expected value of benefit, and a.sub.i.sup.m is the task execution average time;

[0078] The combination of the execution average time a.sub.i.sup.m and the resource r.sub.i.sup.m(t) obtained by comprehensive benefit maximization of the cooperative computing system CS is substituted into Φ(Ω(t)), and Φ(Ω(t)) is minimized to obtain an optimized task dispatching strategy:

[00026] $a_i^m(t) = \{$ $A_i^m(t), \quad Q_i^m(t) \le Y_i^m(t)$ / $0, \quad$ other [0079] when a real queue Q.sub.i.sup.m(t) is not longer than the constant variable transposition queue, receiving, by the manufacturing node, all requests of the tasks in class m that reach at time t, otherwise waiting.

Claims

1-**6**. (canceled)

7. A task-level cooperative optimization dispatching method supporting multi-cloud disconnection disaster recovery, characterized by comprising the following steps: assigning, by a manufacturing node, a task to other matched manufacturing nodes through an end-to-end communication channel not dependent on a service; conducting cooperative optimization dispatching for the task between the manufacturing node and other manufacturing nodes by a CoopDispatch method; synchronizing task processing results of the manufacturing node and other manufacturing nodes among the manufacturing nodes by a LocalAny mechanism, wherein the CoopDispatch method calculates a resource-limited minimum task execution average time in a cooperative computing system composed of a plurality of manufacturing nodes to cooperatively dispatch the task to improve the overall revenue of the cooperative computing system, comprising the following steps: 1) dividing tasks in the cooperative computing system CS into class |M|; representing a task set as Task; for the tasks in class m, m∈|M|, and processing the tasks only in the task set Task.sub.m∈Task where resource R.sub.i exists; for the current manufacturing node, an arrival rate A.sub.i.sup.m(t) of a request for the tasks in class m to other manufacturing nodes at time t satisfies:

$0 \le A_i^m(t) \le A_m^{max} *$ $_{\{T_i \in \text{Task}_m\}}$ wherein A.sub.m.sup.max represents a maximum arrival rate of the tasks in class m, i represents an ith task of the tasks in class m, τ.sub.

{T.sub.i.sub.∈Task.sub.m.sub.} is a flag bit for representing whether the task T.sub.i that can be processed by the resource R.sub.i is in the task set Task.sub.m, i.e.:

$_{\{T_i \in \text{Task}_m\}}$ = { $1, \forall_i \in \text{Task}_m$ / $0, \forall_i$ .Math. Task_m the manufacturing node processes multiple tasks at the same time, the tasks in class m at time t are stored in a queue Q.sub.i.sup.m(t), and all task queues {Q.sub.i.sup.m(t), r.sub.i∈R.sub.i, Task.sub.m∈Task} constitute the following queue matrix Q(t):

$$Q(t) = \begin{bmatrix} Q_1^1(t) & Q_1^2(t) & \text{.Math.} & Q_1^{\text{.Math. } M \text{ .Math.}}(t) \\ Q_2^1(t) & Q_2^2(t) & \text{.Math.} & Q_2^{\text{.Math. } M \text{ .Math.}}(t) \\ \text{.Math.} & \text{.Math.} & \text{.Math.} & \text{.Math.} \\ Q_I^1(t) & Q_I^2(t) & \text{.Math.} & Q_I^{\text{.Math. } M \text{ .Math.}}(t) \end{bmatrix}$$

r.sub.i represents a resource in resource set R.sub.i, and Q.sub.I.sup.|M|(t) represents a last task of the processed tasks in class m with a queue waiting number of |M|; Q.sub.i.sup.m(t) represents a task in class m processed by the resource r.sub.i, i=1 . . . . I; the number of requests of the task in class m received at time t is defined as a.sub.i.sup.m(t), which satisfies: $0 \le a_i^m(t) \le A_i^m(t) * \quad t$ wherein Δt represents a time interval, which is a constant; the task execution average time is as follows:

$a_i^m = \lim_{t \text{ .fwdarw. } \infty} \frac{1}{t} \text{.Math.}_{=0}^{t=1} \quad E\{a_i^m(\quad_{\{T_i \in \text{Task}_m\}})\}$ wherein α is an empirical constant, E{ } represents an expectation, which indicates the expectation of the task execution average time a.sub.i.sup.m; 2) assigning, by the cooperative computing system CS, resources required by the tasks to process the tasks, and assigning the resource CS.sub.i corresponding to the ith task to the resource requested by the task in class m, r.sub.i.sup.m(t): $\text{.Math.}_m \; r_i^m(t) \le R_i$ updating a task processing queue by the manufacturing node through the following model:

$Q_i^m(t+1) = \max(Q_i^m(t) - r_i^m(t) * \quad_m)$ wherein ϑ.sub.m represents the number of the tasks in class m that can be processed by unit resource in a set cycle; 3) constructing a comprehensive benefit model according to the task execution average time a.sub.i.sup.m and the resource r.sub.i.sup.m(t), and obtaining a combination of minimum execution average time a.sub.i.sup.m and resource r.sub.i.sup.m(t) by comprehensive benefit maximization of the collaborative computing system CS to achieve cooperative optimization dispatching, wherein: in step 3), achieving cooperative optimization dispatching by comprehensive benefit maximization of the collaborative computing system CS comprises the following steps: (1) when the task of the manufacturing node is received by other manufacturing nodes, the manufacturing node obtains the following benefits: $\int 1^{(a_i^m)} = \log(1 + \quad_m * a_i^m)$ wherein § 1.sup.(a.sup.i.sup.m.sup.) represents the first part of benefit, β.sub.m is a request revenue constant of the task in class m and a.sub.i.sup.m is the task execution average time; (2) the number n.sub.m(t) of requests of the task in class m processed by the manufacturing node at time t is: $n_m(t) = \text{.Math.}_i \; r_i^m(t) * \quad_m$ ϑ.sub.m represents the number of the tasks in class m that can be processed by unit resource in a set cycle; corresponding revenue is: $\int 2^{(n_m)} = \quad_m * n_m$ wherein n.sub.m is a time expectation of n.sub.m(t), and θ.sub.m is a corresponding revenue constant; § 2.sup.(n.sup.m.sup.) represents the second part of benefit, which depends on the number of served requests; (3) the comprehensive benefit model is: $B = \text{.Math.}_{i,m} \int 1^{(a_i^m)} + \text{.Math.}_m \int 2^{(n_m)}$ wherein i represents the ith task of the tasks in class m; and (4) to maximize the comprehensive benefit to achieve cooperative optimization, a.sub.i.sup.m obtained is the minimum execution average time when B reaches a maximum value, wherein: synchronizing task processing results of the manufacturing node and other manufacturing nodes among the manufacturing nodes by a LocalAny mechanism comprises the following steps: when the manufacturing node receives the tasks of other nodes, with the first part of benefit § 1.sup.(a.sup.i.sup.m.sup.) being nonlinear, adding a transform constant variable y.sub.i.sup.m(t) and a

constant variable transposition queue {Y.sub.i.sup.m(t), ∀i, m} to convert task optimization dispatching into a linear problem:

((t)) = .Math. $E\{Q_i^m(t) * w_m * a_i^m(t) - Y_i^m(t) * \partial_m * a_i^m(t) \mid (t)\}$ wherein w.sub.m and ∂.sub.m are constant variables, Φ(Ω(t)) represents a maximum comprehensive benefit value, Ω(t) represents a difference between a current comprehensive benefit and an average comprehensive benefit, E{ } represents an expectation which here represents an expected value of benefit, and a.sub.i.sup.m is the task execution average time; substituting the combination of the execution average time a.sub.i.sup.m and the resource r.sub.i.sup.m(t) obtained by comprehensive benefit maximization of the cooperative computing system CS in step 3) into Φ(Ω(t)), and minimizing Φ(Ω(t)) to obtain an optimized task dispatching strategy; $a_i^m(t) = \begin{cases} A_i^m(t), & Q_i^m(t) \le Y_i^m(t) \\ 0, & \text{other} \end{cases}$ when a real queue Q.sub.i.sup.m(t) is not longer than the constant variable transposition queue, receiving, by the manufacturing node, all requests of the tasks in class m that reach at time t, otherwise waiting.

8. The task-level cooperative optimization dispatching method supporting multi-cloud disconnection disaster recovery according to claim 7, characterized in that: the manufacturing node is a client of an edge side for workshops of manufacturing enterprises to carry a production management and control system in a multi-cloud environment.

9. The task-level cooperative optimization dispatching method supporting multi-cloud disconnection disaster recovery according to claim 7, characterized in that: the multi-cloud comprises a cloud formed after cloudification of an enterprise management and control business and a sub-cloud of a production workshop of each manufacturing enterprise.

10. The task-level cooperative optimization dispatching method supporting multi-cloud disconnection disaster recovery according to claim 7, characterized in that: in step 3), achieving cooperative optimization dispatching by comprehensive benefit maximization of the collaborative computing system CS comprises the following steps: (1) when the task of the manufacturing node is received by other manufacturing nodes, the manufacturing node obtains the following benefits: $\int 1^{(a_i^m)} = \log(1 + \quad_m * a_i^m)$ wherein § 1.sup.(a.sup.i.sup.m.sup.) represents the first part of benefit, β.sub.m is a request revenue constant of the task in class m and a.sub.i.sup.m is the task execution average time; (2) the number n.sub.m(t) of requests of the task in class m processed by the manufacturing node at time t is: $n_m(t) =$ .Math. $r_i^m(t) * \quad_m$ ε.sub.m represents the number of the tasks in class m that can be processed by unit resource in a set cycle; corresponding revenue is: $\int 2^{(n_m)} = \quad_m * n_m$ wherein n.sub.m is a time expectation of n.sub.m(t), and θ.sub.m is a corresponding revenue constant; § 2.sup.(n.sup.m.sup.) represents the second part of benefit, which depends on the number of served requests; (3) the comprehensive benefit model is: $B =$ .Math. $\int 1^{(a_i^m)} +$ .Math. $\int 2^{(n_m)}$ wherein i represents the ith task of the tasks in class m; (4) to maximize the comprehensive benefit to achieve cooperative optimization, a.sub.i.sup.m obtained is the minimum execution average time when B reaches a maximum value.