International Journal of Grid and Distributed Computing
Vol. 10, No. 1 (2017), pp.149-164
http://dx.doi.org/10.14257/ijgdc.2017.10.1.14

Research on an Improved Grid Task Scheduling Model Algorithm

Liu Feng

Heilongjiang University of Technology,
JiXi 158100, china,
Liufeng8038@163.com

Abstract

On the basis of analyzing the current status and the key technology of grid workflow scheduling, deep research on grid workflow scheduling algorithm under the restraint of time QOS and trust QOS is done in the paper. A grid workflow task scheduling algorithm (GWTS) based on critical tasks under the constraints of trust is designed. Firstly, backward depth of tasks is calculated in GWTS, and critical tasks are ascertained according to execution time on candidate resource. And secondly, the trust of grid resource is computed based on direct experience and recommendation experience synthetically. Finally, tasks are scheduled by decreasing backward depth and resources are closed to meet integrated function of execution time and trust and allocated for critical tasks as a priority. Experiments show that the workflow completion time is reduced and the success rate of task execution is increased by 6-15% and the GWTS algorithm can effectively guarantee grid scheduling resource optimization and improve the scheduling efficiency.

Keywords: *Grid, workflow scheduling, Trust QOS*

1. Introduction

In grid computing, task management, task scheduling and resource management are three necessary basic functions of grid. Effective job scheduling of workflow is a key and also a complete issue to grid system [1-6]. Due to the dynamism and autonomy of grid, plentiful unreliable resources exist in grid environment. How to choose proper resources to meet user demand is a significant and complicated job and thus the scheduling of dependable resources becomes of very importance [7-10].

Grid workflow is an important part in the grid computing environment [11-13]. So far it's mainly applied in scientific research field. Workflow task scheduling not only distributes relative sub tasks to suitable grid resources but also considers dependence relation between different tasks and the execution sequence of them among different resources. As the limited degree of grid workflow ending time is varied, workflow task is divided into two types [14-16]. For the workflow with highly restricted time range, it should ensure that the whole workflow completes in certain deadline. During the scheduling of workflow tasks, the completion time of key tasks determines the operational time of the entire workflow. If it needs reduce time of workflow, key jobs should be preferentially scheduled [17-18].

Grid dynamism and self-governing allows lots of unreliable resources in the grid system. Selecting appropriate resources to complete task scheduling becomes a complex and important job. If key jobs in the workflow are allocated to unreliable resources, it would lead to rescheduling; as a consequence [19-21], it would give rise to extra expenditures, waste of system resources and worse scheduling efficiency, along with unnecessary loss. Under this circumstance, it's rather important to introduce trust to the study on grid scheduling mechanism [22-26].

ISSN: 2005-4262 IJGDC

To solve problem as above, the paper proposes grid workflow task scheduling algorithm based on trust constraint. The algorithm firstly determines the reverse depth of task as to decide the execution order of workflow by processor; then, with reference to trust degree of resource and time of the sub task executed on the resource, it decides resources firstly for key tasks; next, allocate resources to non-key tasks.

2. Grid Workflow

Workflow is a kind of operating process which can carry out totally or partially automatically. With a series of rules followed, document, task or information can be passed and executed among different executors, making the process work automatically and collaboratively as to improve working efficiency [27-28]. The concept of grid workflow was raised on the foundation of business workflow, an automatic execution process of grid task. Workflow management system can wholly define administrative workflow and implement workflow in proper order according to the workflow logic predefined in the computer. Again we say grid workflow is an automatic execution process of grid task.

2.1 Architecture and Execution of Grid Workflow

The main functions of grid workflow management system are divided into two stages. Firstly, it is to define the workflow task and to model the workflow. Secondly, the workflow application is processed, including the implementation of the management workflow and the grid resource interaction. It is shown in Figure1.

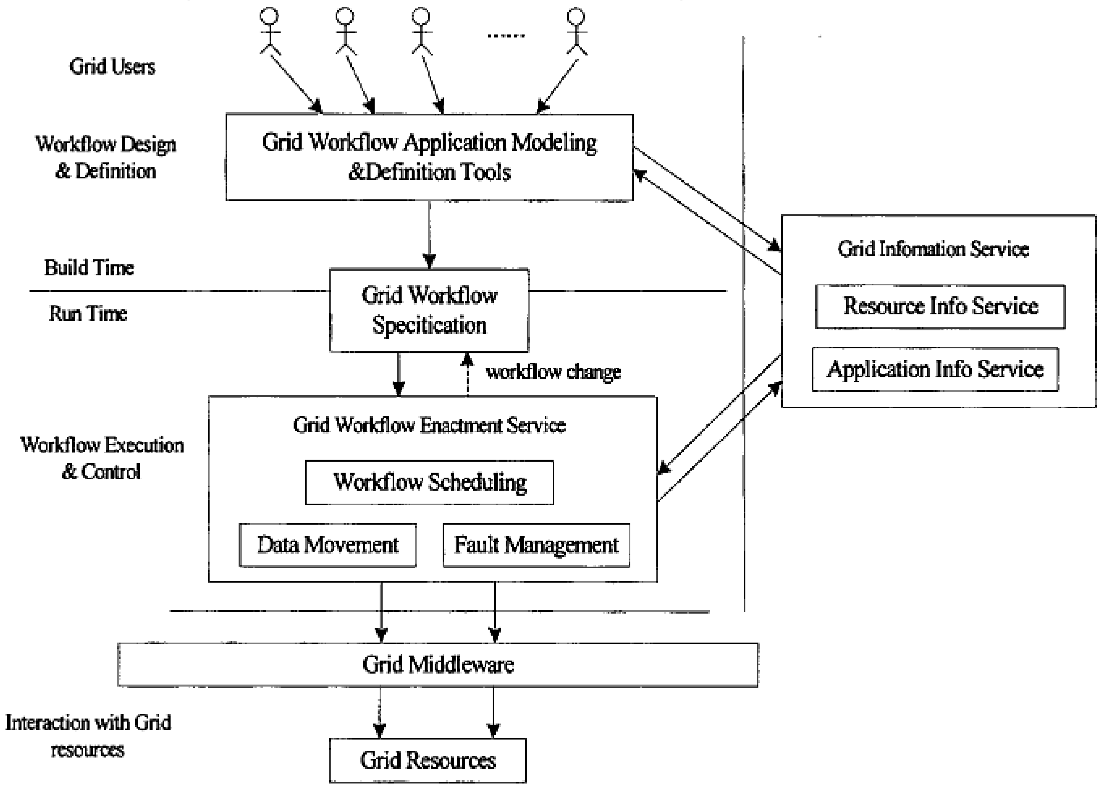


Figure 1. Grid Workflow Management System

The basic process of grid workflow execution includes

(1)End - grid users submit service requests and related requirements.

(2)Grid workflow engine, workflow system uses the appropriate grid workflow description language to convert the service request into metadata.

(3) To decompose the service request from top to bottom, define the task, including the function of the task, the input and output, and generated related data files.

(4) According to the data file select service components, forms logical workflow

(5) According to the current environment and task requirements, to select the appropriate resources for the task, to determine the order of the task.

(6) Perform a task, and pass the final result or data to the user.

2.2Task type of Workflow

According to the definition of Workflow Management Coalition WMFC, the execution structure of workflow task is mainly composed of the following basic types

1Order relation

Task v_1, v_2, v_3 V1, V2 followed by sequential execution. When the task v_1 to perform after the execution of the task v_2. It is shown in Figure2 (a).

2 Parallel relation

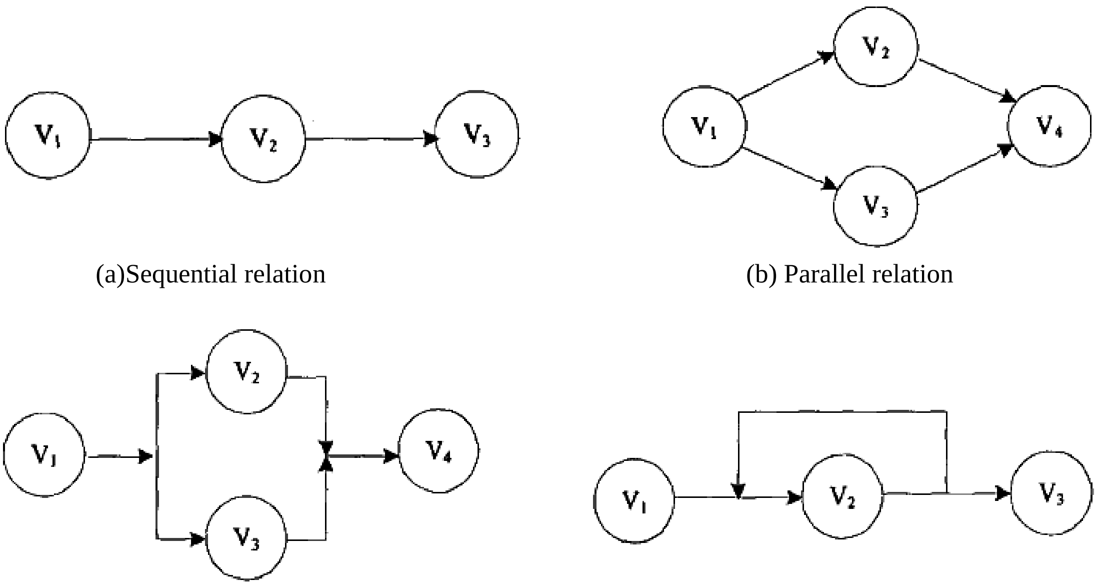
Task v_2 and v_3 parallel execution. That is, these two tasks can be executed simultaneously or in any order. It is shown in Figure2 (b).

3Choice relation

When the task v_1 is executed, the task is to select one of v_2 and v_3 to perform the task. It is shown in Figure2 (c).

4Cyclic relation

Indicates that a task is executed several times, and the task v_2 in Figure 4 can be executed once or more times. It is shown in Figure2 (d).



(a)Sequential relation (b) Parallel relation

(c) Choice relation (d) Circular relation

Figure 2. Four Kinds of Execution Structure of Workflow

3. Description of Problem

3.1. Task Model

Workflow task TG is expressed with V, E, W, C; where V is collection of task nodes v_j ; E is collection of relationship between tasks; e_{ij} is dependence relation between v_i and v_j ; v_i is precursor of v_j ; $v_i \in pre(v_j)$, $succ(v_i)$ means precursor's task collection of v_i ; $v_j \in succ(v_i)$ means task v_i 's subsequent task collection; W is collection task computing expense; weight w_j is computing expense of task v_j ; C is collection of communication time; c_{ij} is time for data transferring from task vi to task v_j ; when v_i and v_j are at the same resource node, c_{ij} =0.

Entry task V_{entry} means task with no precursor; exit task V_{exit} means task with no successor [29].

Here we discuss only tasks of computation-intensive type. For them the communication time is calculated little, which can be neglected.

3.2. Resource Model

The distributed heterogeneous resources in grid environment such as processor's computing resource, storer's memory resource belong to different grid nodes, for which management strategies are diversified, however standard and uniform interface is provide. Resources in the grid environment include the collection of different static attributes and dynamic attributes. Static attributes like CPU, number of processor, geographical position of resource, memory size, memory capacity and network bandwidth etc; dynamic attributes like available number of processing units, current CPU loading, available memory and current storage space [30].

Suppose grid contains resource nodes $m_1, m_2, ..., m_z$ of z grids. Each node mi includes computing resource, storage resource etc. One resource performs higher indicating that its execution time on the resource becomes shorter, otherwise, the execution time is longer. GR is collection of grid resource node mi.

3.3. Scheduling Principle

Grid scheduling distributes tasks to the most suitable resource nodes for execution by certain strategies in the case of meeting QOS such as time and trust constraints. Grid workflow scheduling considers not only task's dependence relationship and also resourceful reliability. The proposed grid workflow task scheduling algorithm based on trust restriction has principles as follows:

1 Subtask priority principle

Owing to the time sequence dependence relationship between subtasks, its scheduling and execution will affect the completion of following tasks. Non-branch tasks have little effect on execution of subsequent tasks. Even if it delays some time, there won't have impacts on the execution of the whole grid workflow.

After the completion of the sub task v_1, due to the branch v_2 will affect the v_4, v_5 task and scheduling, so give priority to the allocation of resources for the v_2, and then for the task v_3 allocation of resources, so that it can improve the task v_3, v_4, v_5 execution parallelism, shorten the scheduling time. In the specific scheduling algorithm, the inverse depth of the task is reflected by the reverse depth, the greater the reverse depth, the more tasks to participate in the branch, the priority in scheduling. It is shown in Figure3.

2 Key path minimization principles

Key path is the workflow path with biggest time consumption in the grid workflow task chart, i.e. from workflow entry task to exit task, the biggest path of summed task computing expense and communication cost is named CP (Critical Path). Task on key path is called key task. The execution time of task on key path plays a decisive role in the scheduling time of the entire grid workflow. To shorten completion time of the whole application, it requires possibly reducing execution time of key path which is the best and consumes the most.

We take Figure 3 for instance. From task v_2 to v_6, there are two paths. Assume v_4's execution time is far less than v_5. There are two candidate resources m_1 and m_2 which have different performance, where m_1's performance is higher than m_2. To minimize v_5's execution time, dispatching resource m_1 to execute v_5 and m_2 to execute v_4 takes shorter time than m_1 matching v_4 and m_2 matching v_5, obviously the scheduling efficiency increased.

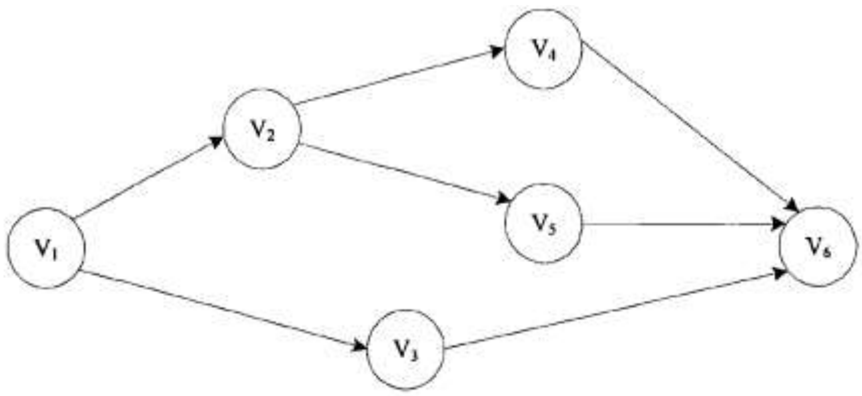


Figure 3.Grid Workflow Application Example

3 Reliable resource prior scheduling principle

If key path is scheduled to resource with better performance but lower trust, grid task may not be timely executed as a result of resource failure or dynamic quit of resource, thus causing re-scheduling, extending scheduling time of key task and raising workflow completion time. In this, it's necessary to choose the resource which has the highest comprehensive beneficial function of execution time and credibility for scheduling.

4 Minimum time and load balancing principle of sequential task

In the premise of meeting grid workflow task QOS requirement, choose for key task the resource which has the best performance and trust for treatment; while for non-key task, choose dependable idle resource to enhance system resource utilization and achieve load balancing.

4. Grid Workflow Task Scheduling Algorithm

At present, there are many workflow task scheduling algorithms based on grid system. MCP and DLS are list scheduling methods in the isomorphic computing environment; HEFT and CPOP are list scheduling methods based on the earliest completion time and key path in isomerous computing environment. Those techniques consider merely task's time QOS, no account of uncertainties, fraudulence of resource nodes, unable to cater to user's service quality requirements in scheduling length and resource confidence. Hence the paper proposed grid workflow task scheduling algorithm, GWTS in short. The main idea of this algorithm is scheduling workflow task from big to small based on task's inverse depth to conform to task dependence relationship and distribute to task the resource which suffices comprehensive function of completion time and trust till all tasks fulfill resource allocation.

The algorithm is investigated on the basis of following grid environment and prerequisites:

(1) GR Description of resource model being met; each resource is mutually connected and can inter-communicate;

(2) Schedule the workflow task of computation-intensive type; describe formally workflow task being met and assume task computing time is greatly bigger than communication time; so communication time is ignored;

(3) Only when precursor task is completely finished and relative program data transferring is over, can the following task start;

(4) One node at one moment can execute one subtask; grid resource is not reliable, may get invalid; resource trust being higher suggests failure rate is lower.

Step 1: for the application of one workflow, from exit task V_{exit} to entry task V_{entry}, calculate inverse depth of each task; the specific calculation method is defined like 2:

Definition 1: task forward depth $TD(j)$ means the biggest path length from entry task V_{entry} to task v_j.

$$TD(j) = \begin{cases} 0, if\,(v_j \in V_{entry}) \\ \max\limits_{v_i \in pre(v_j)} \{TD(i)\} + 1, otherwise \end{cases} \quad (1)$$

Definition 2: task backward depth $BD(i)$ means the biggest path length from task v_i to exit task V_{exit}.

$$BD(i) = \begin{cases} 0, if\,(v_i \in V_{exit}) \\ \max\limits_{v_j \in succ(v_i)} \{BD(j)\} + 1, otherwise \end{cases} \quad (2)$$

The task depth of grid workflow instance in Figure 3 is shown in table 1.

Table 1. Calculation Results of Two Kinds of Hierarchical Lists in an Example

TD	Top Level	BD	Bottom Level
0	v_1	3	v_1
1	v_2, v_3	4	v_2
2	v_4, v_5	1	v_3, v_4, v_5
3	v_6	0	v_6

Step 2: use formula 3 to calculate the average execution time on according candidate resource for each workflow subtask; according to definition and computing method in

equation 5, find out the task on the path with the longest completion time from entry node to exit node; decide key task, i.e. CT.

Definition 3: task execution time is consumed by task t_j for relative candidate resource m_k in the process from starting to ending, expressed by t_{kj}; execution time is related with the resource type and performance, which can be predicted before scheduling and is known; if resource doesn't have problem or becomes ineffective, the execution time is relatively fixed.

Definition 4: task average execution time $\overline{t_j}$ is the mean value of execution time t_{kj} by task in candidate resource collection. It is shown in the formula 3.

$$\overline{t_j} = \frac{\sum t_{jk}}{N(qi)} \quad (3)$$

Definition 5: resource earliest available time avai $[m_k]$ is the earliest idle usable time by resource m_k for scheduling task to the resource, which is also the earliest time for submitting task when or certain time before local task is done.

Definition 6: task beginning time est(v_j) is the earliest starting time of task v_j on processing machine m_k ; the time is the biggest value of summation of the earliest available time avai $[m_k]$ of task's node and completion time and communication time c_{ij} of precursor task.

$$est(v_j) = \max\{avail[m_k], \max_{v_i \in pre(v_i)} (etc(v_i) + c_{ij})\} \quad (4)$$

Definition 7: task completion time $est(v_j)$ refers to the sum of task starting time $est(v_j)$ and task's execution time t_{kj} on resource. Comprehensive formula 4 and 5 can get the formula 6.

$$est(v_j) = est(v_j) + t_{kj} \quad (5)$$

$$est(v_j) = \max\{avail[m_k], \max_{v_i \in pre(v_i)} (etc(v_i) + c_{ij})\} + est(v_j) = est(v_j) + t_{kj} \quad (6)$$

Step 3: compute resource credibility: resource credibility includes direct confidence and recommendation confidence; direct confidence is associated with resource successful execution times, failure times, delay times and time decaying function, which is calculated as formula 5; recommendation confidence, the direct confidence of recommendation resource itself as well as recommendation factor are connected with time decaying function, which is acquired by formula 6; make direct confidence impact factor α mean the weight taken up by the direct confidence degree; generally speaking, the influence of direct trust degree is biggest than recommendation confidence. In different grid environments, α varies along.

$$\Omega(m,t,c) = \alpha * D(m_j,t,c) + (1-\alpha) * R(m_j,t,c) \quad (7)$$

Step 4: according to task backward depth from big to small, distribute firstly key task, with task execution time and resource credibility as comprehensive beneficial function to allocate resources for all workflow tasks.

Here it discusses primarily computing-intensive task, whose communication time is less than computing time and thus can be neglected.

The objective of scheduling in the paper is to choose $\min(f(v_j))$ resource by using time and trust as target function $f(v_j)$ calculated in the expression as shown in 9. λ means the weight of task scheduling; the bigger the λ is, the greater influence the trust has on task scheduling; trs_j is loss caused by the trust, which is acquired by formula 8; $\{1-\Omega(m_j,t,c)\}$ represents the risk probability of resource; bigger trust value means $\{1-\Omega(m_j,t,c)\}$ is smaller, implying that failure has less impact on task scheduling. GWTS algorithm firstly selects for key task the resource which has minimum comprehensive function of execution time and trust; then according to the scale of task $BD(j)$ from big to small, selects for non-key task the resource which has minimum comprehensive function of completion time and trust.

$$trs_j = t_{jk} * \{1-\Omega(m_j,t,c)\} \quad (8)$$

$$f(v_j) = \begin{cases} \lambda * trs_j + (1-\lambda) * t_{jk}, if(v_j \in CT) \\ \lambda * trs_j + (1-\lambda) * ect(v_j), if(v_j \notin CT) \end{cases} \quad (9)$$

To sum up, the grid workflow task scheduling algorithm based on critical path under trust constraints is as follows:

Algorithm: Scheduling task on resource using GWTS

Input: TG,GR

Output: allocated m_n for each task v_j

Begin

1. For each task v_j from V_{entry} to V_{exit} in TG
2. Calculate TD(j) using(1)
3. End for
4. For each resource m_j
5. Calculate $\Omega(m_j)$ using(7)
6. End for
7. From V_{entry} to V_{exit} in TG
8. Calculate $ect(v_j)$ for each task using(2)
9. Find CP according to $ect(v_{exit})$ and add tasks to CT
10. Rank task according to TD(j)
11. While there is task not allocated
12. Select v_j which has the minimum TD(j)
13. Calculate trs_j using(9) for each resource in Q(j)
14. Calculate $f(v_j)$ using(10)
15. Select m_n for task v_j having min $f(v_j)$
16. End while
17. End

5. Experiment Design and Discussion

5.1. Grid Simulation Tool

Currently, the commonly used grid simulator mainly has SimGrid, GridSim, MicroGrid, Bricks and other simulation tools.

1 SimGrid

SimGrid is the research and development of Santiago grid research and Innovation Laboratory of California University in the United States. The distributed parallel scheduling is mainly provided with suitable models and abstractions, and accurate simulation results are provided. SimGrid includes two versions, one is SG, suitable for simulation based on DAG centralized scheduling, to provide the underlying API to build simulation environment. The other is MSG, based on SG, which provides the application oriented API, which is suitable for simulating the scheduling of multiple independent tasks.

2 GridSim

Gridsim is a Java based grid simulation tool developed by Melbourne University in Australia, Rajkumar, in order to achieve resource allocation of computational economy model.

It provides rich library functions as well as different parameter configurations for heterogeneous distribution of grid entities, such as resources, applications, user, scheduler and user agent etc.

GridSim can simulate the global computing, network resources, provide tasks to find, the task of virtual processing and other functions. GridSim uses a layered approach to simulate grid task scheduling, focusing on one aspect of the function of each layer.

GridSim tools include multiple entities, such as grid resources, network, information services, and the simulati these entities with the aid of the interface provided by SimJava, communicate by sending and receiving event objects to simulate the interaction behavior of grid bottom layer

This layer includes two layer Gridlet and GridSimRandom grid system model, using the model to achieve scheduler model, grid system. The upper layer is the application layer, which mainly includes the different application scenarios, including the establishment of the simulation experiment, the establishment of the user model and so on.

Resource entities are mainly composed of GridResource and a series of auxiliary classes in the GridSim toolkit. Now, contains many distributed and heterogeneous resources.

The heterogeneity of resources is mainly reflected in the processing speed, the number of processors, the processing cost, the number of machines, the local load parameters, the internal scheduling strategy and so on.

Different parameters can be established SMP, PC, cluster, and other different resource models. The layer also includes behavior of different entities. Such as task submission, resource registration, resource query, task processing, etc.

Service mainly provide to resource registration and resource query function, through the GridInformationService class to achieve. Network entities‘s the input class and output class to achieve, the main simulations data transmission and delay in the network.

3 MicroGrid

MicroGrid is the leading development of California University Santiago parallel system architecture group. By using the existing physical resources to simulate the virtual grid environment. The important part of MicroGrid is the simulation engine, which is built on the basis of the parallel distributed simulation engine. By implementing completely controllable virtual grid environment, it can help us to do the grid system design and performance evaluation of the grid system.

4 GridNet

GridNet is mainly to simulate the dynamic data replication strategies in data grid. GridNet replica decision based on cost estimation model is proposed.

GridNet is based on the modular design. GridNet simulation environment mainly includes the following three types of nodes, clients, servers and cache nodes. The client is responsible for generating the data access request. The server side represents the storage node. The cache node represents an intermediate storage node, which is used to copy the data stored on the server.

5 Bricks

Bricks is the research and development of Tokyo Institute of Science in japan. The main simulation provides remote access to the scientific computing library and client server type of computing system. Bricks is composed of a wide area computing environment and a scheduling unit, which can be used to test various scheduling algorithms. However, centralized global scheduling has certain limitations.

Here Gridsim tool is utilized to construct simulation environment; next the proposed GWTS algorithm is compared with HEFT algorithm and CPOP.

5.2 Introduction of related algorithms

1 HEFT algorithm [31] is classical scheduling algorithm in heterogeneous environment; it divides scheduling process into two phases: weight distribution and task allocation. At first, as per task's execution time and dependence relationship, from entry task to exit task, assign different weights to task; then in the other phase, according to task weight from small to big, allocate task to the node of resource with the earliest ending time;
2 CPOP algorithm [32] considers preemptively key task; assign different weights in accordance to the length from task to exit and entry task; later in task allocation period, do from entry task; take priority to task on key path; assign key task to the node of resource with the shortest execution time; if it's non-key task, distribute task to the node of resource with the earliest ending time.

5.3 Experimental Environment Configuration

The experiment uses GridSim packet as foundation to create grid simulation environment [33], a heterogeneous multi-cluster grid system model. The model contains ten resource sites (CE_1-CE_10). The configuration of each resource site is listed in details in Table2. The experimental environment generates randomly workflow which contains different subtasks.

Table 2. Grid System Model Configuration of Simulation Experiment

Resource name	Number of CPU systems	MIPS
CE_1	128	450
CE_2	64	300
CE_3	256	410
CE_4	64	340
CE_5	512	230
CE_6	256	420
CE_7	128	230
CE_8	128	340
CE_9	64	450
CE_10	256	300

The resources attribute increases the reliability level(R,N,U,M) and the delay and failure rate of the reliability level are shown in Table 3. The performance of the resource is not stable, and the failure probability of the resource is expressed as the failure probability of the resources, which reflects the possibility of the unilateral revocation of the resource.

Table 3. Reliability Level of Resources

Reliability level	Delay rate	failure rate
Reliabel(R)	[0,5]	[0.01,0.1]
Normal(N)	[5.20]	[0.1,1]
Unreliable(U)	[20,50]	[1,10]
Malicious(M)	[50,100]	[10,100]

The influence of direct trust and recommendation trust is equivalent in the initial time, so the confidence factor of formula 7 is α =5. According to the reliability level defined in table2, the experiment sets 2 resource scene, it is shown in table 4.

Table 4. Scene Distribution of Reliable Resources and Non reliable Resources

Scenario	R	N	U	M
SC1(Reliable)	85%	7%	5%	3%
SC2(Unreliable)	40%	30%	20%	10%

Two indicators are used to evaluate the performance of the algorithm.
(1) MakeSpan

Span means time span from task beginning to resource execution till all tasks complete; the shorter time the completion takes, the better performance the algorithm can achieve and the better performance the grid system can get.
(2) Task execution success rate

Workflow execution success rate analyzes the effect of trust QOS on task scheduling, S means the number of successful scheduling; n is total number of scheduled tasks. Calculation method is shown in formula 10.

$$suc\_rate = \frac{\sum_{i=1}^{m} s}{n} \qquad (10)$$

5.4 Experimental Result Analysis

Here a simulation program is compiled to GRIDSIM tool, which includes relative resource description, task description and relevant scheduling algorithm. The experiment sets SCI and SC2 scenes. Adjust the influence of trust on the scheduling according to different value of λ, which is experimentally implemented as below:

(1) In SC1, when $\lambda = 0.5$, Schedule contains 20,50,100,200,500 workflow tasks, the completion time of the different algorithm is shown in Table 5.

Table 5. The Completion Time of the Task of $\lambda = 0.5$ in SC1

Task number	HEFT	CPOP	GWTS
20	170	160	158
50	420	420	400
100	670	680	631
200	1567	1459	1412
500	2644	2645	2356

(2) In SC2,when $\lambda = 0.9$ Scheduling different workflow tasks. The completion time of the algorithm is shown in Table 6.

Table 6. The Completion Time of the Task of $\lambda = 0.9$ in SC2

Task number	HEFT	CPOP	GWTS
20	233	190	160
50	520	500	456
100	850	812	617
200	1756	1765	1570
500	3200	3250	2611

(3) In SC2,when $\lambda = 0.9$ Scheduling different workflow tasks. The success rate of workflow execution is shown in Table 7 and in Figure4.

Table 7. Success Rate of Workflow Execution of $\lambda = 0.9$ in SC2

Task number	HEFT	CPOP	GWTS
20	61	63	66
50	63	67	73
100	70	69	79
200	72	74	80
500	75	80	89

The major shortcoming of HEFT algorithm is inability to reduce scheduled MAKESPAN during each scheduling, because the ready task with the earliest and minimum starting time is not necessarily key task, and task preferentially scheduled to processing resource node would take its own time, as a result key task can't get firstly the resource node with faster processing speed when resource is selected, making MAKESPAN of the whole workflow grow. But HEFT algorithm and CPOP algorithm don't take into account the trustability of resource. The proposed GWTS algorithm made improvements of the two methods, by merging the influence of QOS trust and QOS time on scheduling.

If tasks on key path are distributed to the resource which has the shortest execution time and worst confidence degree, it would lead to re-scheduling due for resource failure. If at this moment tasks are assigned to other candidate resources, key task needs waiting, extending the completion time of the whole workflow. GWTS algorithm considers task's average execution time on each candidate resource when deciding key path. Also by integrating resource trust and execution time, it distributed resource preferentially for key task, shortening the execution time of workflow. Table 7 reveals that the proposed GWTS algorithm considered resource credibility on key path and increased task execution successful rate. More tasks indicates GWTS algorithm gains higher task execution success rate. Through adjustment of the proportion of trust as per the value of λ, the influence of trust on scheduling is well assured.

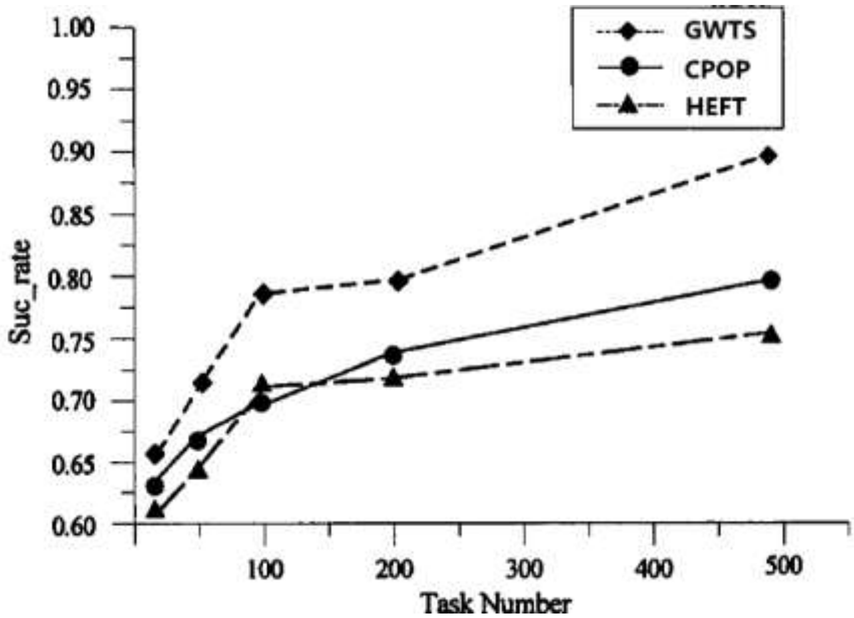


Figure 4. In SC2, when $\lambda = 0.9$ the Success Rate of Workflow Execution

6. Conclusion

This paper proposes a grid workflow task scheduling algorithm in grid environment. The GWTS algorithm first calculates the inverse depth of the task, and then determines the key task as well as the trust degree of the resources. Finally, task scheduling is performed on the execution time and trust degree. In different scenarios, analyzed the performance of GWTS algorithm and other algorithm. Experiments show that, compared with HEFT algorithm and CPOP algorithm, GWTS algorithm reduces the running time of the whole workflow, and improves the successful execution rate of 5-15%.

Acknowledgement

This work was supported by the project of Nature Scientific Foundation of Heilongjiang Province (F2016038).

References

[1] L. Xi, H. Zhigang and Y. Zhaokun, "Research and development of grid workflow scheduling algorithm based on deadline satisfaction", Computer research and development, vol. 5, **(2011)**, pp. 877-884.

[2] D. Bin, "Research on grid workflow task scheduling strategy and algorithm", Science and technology information, vol. 19, **(2011)**, pp. 497-498.

[3] Wu, "Akitoshi cloud workflow service composition and activity scheduling of Strategy Research", HeFei University of Technology, **(2011)**.

[4] P. Xun, "Swarm intelligence and distributed system research on the application", Electronic Science and Technology University, **(2012)**.

[5] L. Xi, "Grid workflow scheduling model algorithm for reliability research of Central", South University, **(2011)**.

[6] Y. Chun, L. Xiaoping, W. Qian and Z. Yi, "Based on reverse layered grid workflow scheduling algorithm", Computer journal, vol. 2, **(2008)**, pp. 282-290.

[7] T. Guozhong, H. Y. Jiong, X. Jian and C. Zhen, "Based on resource state reliability of grid workflow scheduling algorithm", Computer engineering and application, vol. 18, **(2008)**, pp. 115-118.

[8] T. Guozhong, H. Y. Jiong and S. Qingang, "Based on the effective resources of grid workflow task scheduling algorithm", Computer engineering, vol. 11, **(2008)**, pp. 80-82.

[9] L. Jinzhong, J. W. Xia, Z. B. Zeng and C. X. Liu, "Grid workflow scheduling algorithm research: a review", Application Research of computers, vol. 8, **(2009)**, pp. 2816-2820.

[10] L. Yang, "Research on dynamic grid workflow scheduling algorithm based on deadline and adaptive", Huazhong University of Science and Technology, **(2012)**.

[11] L. Haitao, "Research on workflow scheduling method in cloud environment", Beijing Institute of Technology, **(2015)**.

[12] T. Zhenjiang, "Grid workflow scheduling algorithm based on survivability", Harbin Institute of Technology, **(2006)**.

[13] S. Xiang, Y. J. Liu, Y. Jian and Y. Yuanyuan, "Based on Chaos Genetic Algorithm for grid workflow scheduling application", Application Research of computers, vol. 9, **(2013)**, pp. 2645-2647.

[14] L. Gongbo, "Research on grid manufacturing scheduling algorithm based on workflow", South China University of Technology, **(2010)**.

[15] H. Yun, "Grid workflow task scheduling algorithm based on service quality constraints", Xinjiang University, **(2011)**.

[16] W. Mingquan, "Load balancing of grid workflow scheduling algorithm", Xinjiang University, **(2011)**.

[17] Y. Xin, "An adaptive grid workflow scheduling algorithm based on domain trust", Huazhong University of Science and Technology, **(2011)**.

[18] Y. Huan, "Research on grid workflow scheduling algorithm based on budget constraint", Nanjing University, **(2014)**.

[19] W. Dawei and J. Shen, "An improved workflow scheduling algorithm in grid computing", computer technology and development, vol. 2, **(2014)**, pp. 71-75.

[20] Z. Min, Y. Qingsong, H. Jun, Z. Wenjie and Z. Yan, "Based on GAPSO algorithm-based grid workflow scheduling", Computer applications and software, vol. 4, **(2011)**, pp. 236-238-241.

[21] L. Huimin and Y. Zhaokun, "A workflow scheduling algorithm based on task competitiveness", Journal of Henan University (Natural Science Edition), vol. 1, **(2012)**, pp. 87-91.

[22] Cheng, S. Jiyin, L. Linlin, M. Sheng and Z. Xin, "Study on the military grid workflow scheduling algorithm", Computer Engineering, vol. 10, **(2012)**, pp. 45-47.

[23] Y. Zhaokun, H. Zhigang, L. Xi and X. Peng, "Reliability enhanced grid workflow scheduling algorithm under time constraints", Small and micro computer system, vol. 7, **(2012)**, pp. 1528-1532.

[24] Z. Xiaohong, "A grid workflow scheduling algorithm for communication overhead", Journal of Jianghan University (Natural Science Edition), vol. 3, **(2015)**, pp. 278-282.

[25] S. Lili, "A grid workflow dynamic scheduling algorithm", Computer system application, vol. 7, **(2009)**, pp. 45-47.

[26] D. Bin, "Collaborative scheduling strategy of workflow based on OGSA grid", China University of Petroleum (Hua Dong), **(2012)**.

[27] Ding, C. Guoliang and G. Jun, "Unified resource mapping strategy in computational grid environment", Journal of software, **(2008)**, pp. 12-19

[28] A. Ranaweera and D. P. Agrawal, "A scalable task duplication based algorithm for heterogeneous systems", Proc of the ICPP. Los Alamitos: IEEE Computer Society Press, **(2010)**.

[29] L. Weiwei, Q. Dexing and L. Yongjun, "The independent task scheduling in the tree grid computing environment", Journal of software, **(2008)**, pp. 2352-2361

[30] Y. Yuan, L. Xiaoping and W. Qian, "The optimization method of grid workflow time with cost constraints", Computer research and development, **(2009)**, pp. 194-201.

[31] P. Radu and W. Marek, "Bi-criteria scheduling of scientific grid workflow", IEEE transactions on Automation Science and Engineering, **(2012)**, pp. 364-380.

[32] L. Chao, Z. Qiaoming and L. Peifeng, "Review of research on grid workflow scheduling", computer application and software, **(2010)**, pp. 279-283.

[33] C. Haijun and S. Xuanhua, "Efficient and dependable scheduling of DAG workflow job in grid", Spectial issue on High performance computing and communication, **(2012)**, pp. 201-223.

Author

Liu Feng. He received his B.S degree in computer Science from Heilongjiang University of Science and Technology. He received his M.S degree in computer Science from Liaoning Technical University. He is an Associate professor in Heilongjiang University of Technology. His research interests include database and algorithm.