2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science

HIILDPM: A Distributed Resource Pricing Method

Jianzhong Qiao
College of Computer Science and Engineering,
Northeastern University, Shenyang, China
qiaojianzhong@mail.neu.edu.cn

Enze Wang
College of Computer Science and Engineering,
Northeastern University, Shenyang, China
wangenze@mail.neu.edu.cn

Shukuan Lin
College of Computer Science and Engineering,
Northeastern University, Shenyang, China
linshukuan@mail.neu.edu.cn

Abstract—**Using the economic theory to solve the problem of the distributed resource allocation has made a lot of achievements, but how to determine the price is rarely discussed. This paper proposes a distributed resource pricing method (HIILDPM), which takes full account of historical transaction records and individual loads, and gives out the corresponding pricing algorithm. A distributed resource allocation algorithm HIIL-CDA is designed, which the pricing part adopts HLDPM and the distribution part adopts CDA (continuous double auction). The results show that HIILDPM has better price convergence, and HIIL-CDA has higher resource utilization and efficiency.**

Keywords-economic theory, pricing, historical transaction records, individual load, resource allocation

I. INTRODUCTION

It is a new way that uses the concept of market to realize the optimal resources allocation by independent autonomous decision-making for the distributed system environment. Many achievements have been obtained based on the economic theory to solve the problem of distributed resource allocation, but there are some deficiencies in the impact factors of the price, which cannot response to the characteristics of the distributed environment very well, and influences the efficiency of resource allocation. In this paper, we take into account the characteristics of the price factors, and propose a distributed resource pricing method (HIILDPM) based on historical transaction records and individual load, and design the resource allocation algorithm HIIL-CDA which improves the resource utilization and implementation efficiency.

II. RELATED WORKS

Using economic theory to solve the problem of distributed resource allocation, a problem must be faced that is the price. At present, the pricing methods can be divided into three categories: one is the centralization approach oriented to supply and demand, which algorithm ideas derived from Walras equilibrium[1], the ultimate goal is to achieve market supply and demand balance by minimizing the excess demand. Wolski[2] introduced this algorithm into a distributed environment, and made reasonable system assumptions and systematic analysis. This kind of method requires the global information, has poor scalability for distributed environments, the current application is less; the second is the discrete approach based on historical transaction records. This approach first appeared in [3], which the bid is generated asynchronously and affected by other resource quotations, and needs to reference the historical transaction. In [4], each price adjustment is based on the previous price, when the resource utilization rate of the producer is less than the minimum resource utilization (man-made), the price is cut down, otherwise resource price is increased. The consumers use a similar approach to adjust the price. The second method uses the historical transaction information, although the price adjustment parameter is easy to obtain, systematic method is not formed; the third is the discrete approach based on individual rational demand-oriented, which the individual tends to higher individual income to a certain extent by following some self-interest rules. ZI strategy, GD strategy [5], Kaplan strategy[6] adopt the discrete pricing method, which only rely on the changes in individual parameters to adjust the pricing, such as resource costs, task deadlines, and so on.

The pricing method HIILDPM proposed in this paper use historical transaction information and individual load to support participant independent decision-making, belong to the mixed methods of the second and the third. The initial pricing of HIILDPM references the hardware price, the price adjustment references the historical records and individual load. This ensures that the price spread is not very large, and the distribution is efficient.

III. HIILDPM PRICING METHOD

A. Related assumptions and definitions

Assuming there are m resource providers $P=\{P_1, P_2,\ldots,P_m\}$and n resource consumer $C=\{C_1, C_2,\ldots,C_n\}$ in the system, each resource provider has a resource $R=\{R_1, R_2,\ldots,R_m\}$, Each resource consumer i has an unequal number of tasks $Ti = \{Ti1, Ti2, ..., Ti_{Xi}\}$, where Xi represents the number of tasks on consumer i.

The resources on the provider i can be expressed by the quaternion $R_i=\{n_{now\_i}, A_i, P_{init\_i}, Cap_{init\_i}\}$, the pricing A_i of resource i is based on the initial hardware price P_{init_i}, the initial processing capability is Cap_{init_i} , n_{now_i} is the current task queue length. In many studies, there are resource costs, which are closely related to the hardware price[7]. The initial price of P_{init_i} is based on hardware prices:

P_{init_i} = hardware price / estimated working hours (1)

The task j of consumer i can be expressed by the quaternion $T_{ij}=\{n_{now\_i}, B_{ij}, P_{init\_i}, Cap_{init\_i}\}$, similar to A_i, the task bid B_{ij} is based on the P_{init_i}, and the pricing is adjusted by n_{now_i} and Cap_{init_i}. The specific parameters are shown in Table 1.

The price influencing factors include the individual influencing factors and the system influencing factors. The individual influencing factors are the basis that the individual can bid on the goods independently, such as the task deadline, the task execution record, the individual load, etc. The system influencing factors is the basis of the bid that the individual obtains in the process of system running, such as resource supply and demand situation, node load. For the resource providers and consumers, the hardware price is the basis for resource bid, the load is one of the basis of resource price adjustment. Individual load is the length of the task queue on the node or the time consumed by all the tasks in the node execution queue. In this paper, the individual load mainly considers the current task queue length, that is, the number of tasks. Resource types only consider CPU processing capability.

TABLE 1 MODEL PARAMETER DESCRIPTION

Parameter	description
P_{init_i}	node i hardware resource price
B_{ij}	the bid of task j of resource consumer of node i
A_i	resource pricing of resource provider of node i
n_{now_i}, $n_{history\_i}$	n_{now_i} represents the length of the current task queue for participant i， $n_{history\_i}$ represents the number of historical transaction records for node i
Cap_{init}, Cap_{now_i}	Cap_{init_i} represents the processing capability of the single task execution at node i（MIPS）， Cap_{now_i} represents the processing capabilities of node i at the moment
$p_{ex\_p\_i}$, $p_{ex\_c\_i}$	$p_{ex\_p\_i}$ represents the historical transaction price of resource provider node i; $p_{ex\_c\_i}$ represents the he historical transaction price of resource consumer node i
A, β, γ	A represents the final transaction price adjustment parameter， β represents resource price adjustment parameters， γ represents the task price adjustment parameter
TL, RL	TL is a task price adjustment parameter, RL is a resource price adjustment parameter
$Standard_C$	$Standard_C$ represents the standard processing capability of consumer nodes in the current system
$Standard_P$	$Standard_P$ represents the standard processing capability of the provider node in the current system

B. The pricing method

From the transaction point of view, for the resource consumers, the longer the current task queue, the stronger the desire that the task assigned to other nodes, the higher the bid; the stronger their processing capacity, the greater the possibility that they perform tasks, the lower the bid. For the resource providers, the longer the current task queue, the weaker the processing capacity, then the price is lower; the stronger the processing capacity, the better the processing capacity recovery, the greater the bid. Whether a node is a resource consumer or a resource provider, the rules for the initial pricing and the active price adjustment are similar and are affected by their own processing capacity. In order to facilitate the transaction, the initial pricing and the active price adjustment of buyers and sellers in HIILDPM use the same way and parameters, which ensure that the price deviation is not big.

The bid of task k on the node i for resources can be divided into pricing and price adjustment:

$$B_{ik}=\begin{cases} p_{c\_i} & NewConsumer\ or\ Active\ price\ adjustment \\ p_{\text{previous\_i}}+\Delta p_i & else \end{cases} \quad (2)$$

$$p_{c\_i}=\begin{cases} 0 & n_{now\_i}=0 \\ p_{init\_i}*[1+\dfrac{n_{now\_i}}{TL}-\dfrac{1}{RL}(\dfrac{Cap_{init\_i}}{Standard_L}-1)] & else \end{cases} \quad (3)$$

$$\Delta p_i=\begin{cases} 0 & p_{previous\_i}\ge ask(0) \\ \gamma\, p_{ex\_c\_i} & else \end{cases} \quad (4)$$

$$p_{ex\_c\_i}=\begin{cases} \sum_{k=1}^{n_{history\_i}} p_k \Big/ n_{history\_i} & history\ exist \\ SystemResourceUnitPrice & else \end{cases} \quad (5)$$

$$SystemResourceUnitPrice_C=\sum_{j=1}^{n_C} p_{init\_j} / n_C \quad (6)$$

When new resource consumers join the system, the initial price is calculated using formula (2) and formula (3). Similarly, the task uses the above calculation to adjust the price. The resource consumers of the system adjust the price using formula (2), (4), (5) and (6).

When some resources and tasks match successfully, the prices of the residual resources and tasks which do not matched is affected, this is equivalent to a new system initial state. Therefore, the initial price calculation and the active price adjustment can use the same calculation. In formula (2), $P_{previous\_i}$ represents the bid of the resource required by the consumer i at the end of the last calculation. P_{c_i} in formula (3) represents the initial bid value of resource consumer i for the resource required by the task k, which is based on *Pinit_i*. n_{now_i}/TL indicates that the effect of the current task queue length on the price. When n_{now_i} is zero, there is no task on the resource consumer node, and the task price is zero. The greater the value of n_{now_i}, the heavier the load of the current resource consumer, the greater the possibility that the task will be handed over to other nodes, the higher the bid of the resource required for the task, the easier transaction will happen. So the TL correlation is positive. *Cap / StandardL* represents the relationship between the processing capability of the current consumer and the minimum resource processing capability of the system. When *Cap* equals to *StandardL*, the processing capability has no effect on the price. The *Cap / Standard* value is not less than 1. The stronger the processing capability of the current resource consumer node, the stronger the ability that recovers normal from the overloaded state, the higher the probability that executes the task, the lower the bid of the resource required for the task, the less the transaction happens. So the *RL* correlation is negative.

If the resources and the tasks are not match, the price needs to be adjusted passively. If the task of the resource consumer has never succeeded in matching with the resource, there is no historical transaction record on the node, and the average hardware price of the current system resource consumer in formula (6) is used as the price adjustment. At the same time, the task price adjustment parameter γ is used to further correct the price adjustment. The n_C in formula (6) represents the number of the resource consumers in the current system. If the resource consumer has a historical transaction record, the price adjustment is calculated using the formula (5). The P_k in the formula (5) represents the transaction price of the task in the k-th historical transaction record. In formula (4), when the task bid $P_{previous\_i}$ at the end of the previous round is not less than the minimum resource price ask(0) in the system, it means that at least there is a resource in the system which has a lower price than the bid of current resource consumer. Although this match is not successful, the task can be allocated in theory, the price needs not to be adjusted passively. Otherwise, the price needs to be adjusted passively. The algorithm that resource consumers of the system adjust the bid is shown as algorithm 1.

Algorithm 1 Task k Bid Algorithm of the resource consumer i

1. BEGIN
2. IF the resource consumer i is the new the consumer THEN
3. $B_{ik}=p_{init\_i}*[1+\frac{n_{now\_i}}{TL}-\frac{1}{RL}(\frac{Cap_{init\_i}}{Standard_L}-1)]$
4. ELSE IF $P_{previous\_i}$>= ask(0) THEN //ask(0) is the minimum pricing of the resources in the system
5. $B_{ik}=P_{previous\_i}$
6. ELSE IF a transaction occurred on the resource consumer i THEN
7. $B_{ik}=P_{previous\_i}+\gamma\sum_{k=1}^{n_{history\_i}} p_k \Big/ n_{history\_i}$
8. ELSE
9. $B_{ik}=P_{previous\_i}+\gamma\sum_{j=1}^{n_C} p_{init\_j}/n_C$
10. END IF
11. END IF
12. END IF
13. END

Similarly, the pricing method of the resource provider i can be expressed as:

$$A_i=\begin{cases} p_{p\_i} & NewProvider\ or\ Active\ price\ adjustment \\ p_{\text{previous}\_i}-\Delta p_i & else \end{cases} \tag{7}$$

$$p_{p\_i}=\begin{cases} p_{init\_i} & n_{now\_i}=0 \\ p_{init\_i}*[1-\frac{1}{RL}(\frac{Cap_{init\_i}}{Standard_L}-1)+\frac{n_{now\_i}}{TL}] & else \end{cases} \tag{8}$$

$$\Delta p=\begin{cases} 0 & p_{previous\_i}\le bid(0) \\ \beta p_{ex\_p\_i} & else \end{cases} \tag{9}$$

$$p_{ex\_p\_i}=\begin{cases} \sum_{k=1}^{n_{history\_i}} p_k \Big/ n_{history\_i} & history\ exist \\ SystemResourceUnitPrice_P & else \end{cases} \tag{10}$$

$$SystemResourceUnitPrice_P=\sum_{j=1}^{n_P} p_{init\_j}/n_P \tag{11}$$

The resource providers and the resource consumers calculate the price at the same pricing concept, which ensure that the price gap will not be too large and easy to trade and adjust price. But there are some differences, in formula (7), the resource provider use price reduction to improve the success rate of resource transactions in the process of passive price adjustment. In formula (9), when the current resource cannot match any resource consumer for the task bid, and the resource price is not greater than the maximum task price B (0), it means that there is at least one resource consumer whose bid for the task is higher than the price of current resource in the system. Although this match is not successful, but the resources can be matched theoretically, the price needs not to be adjusted passively. Otherwise the price needs to be adjusted passively using formula (9) to (11). The resource pricing algorithm of the resource provider i is shown as algorithm 2.

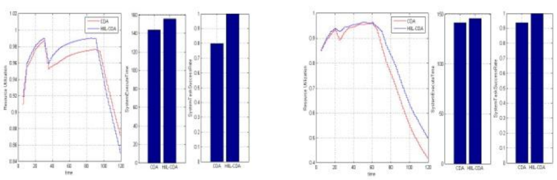
Algorithm 2 Resource Pricing Algorithm of the resource provider i

1. BEGIN
2. IF resource provider i is a new provider THEN
3. $A_i=p_{init\_i}*[1-\frac{1}{RL}(\frac{Cap_{init\_i}}{Standard_L}-1)+\frac{n_{now\_i}}{TL}]$
4. ELSE IF $P_{previous\_}$<=bid(0) THEN //bid(0) is the minimum bid of theresources in the system
5. $A_i=P_{previous\_i}$
6. ELSE IF a transaction occurred on the resource provider i THEN
7. $A_i=P_{previous\_i}-\beta\sum_{k=1}^{n_{history\_i}} p_k \Big/ n_{history\_i}$
8. ELSE
9. $A_i=P_{previous\_i}-\beta\sum_{j=1}^{n_P} p_{init\_j}/n_P$
10. END IF
11. END IF
12. END IF
13. END

IV. EXPERIMENTAL ANALYSIS

In order to illustrate the effectiveness of the proposed algorithm HIILDPM, we design HIIL-CDA algorithm which the pricing part adopts HLDPM and the distribution part adopts CDA matching rule. Then HIIL-CDA is compared with ZI-CDA (The pricing part adopts ZI[8] and the distribution part adopts CDA matching rule) and CDA. The simulation experiment is based on a self-developed toolkit called EDistributed, which is designed based on GridSim[9] to simulate the application of economic theory to solve the problem of distributed resource allocation.

If the number of the resources is less than the number of the tasks, it is very likely that the bid and the ask queues do not have the crossover point but can match, and the task that does not match can meet the matching condition. In the experiment, a *BadLimit* is set, When the number of times exceeds this parameter, the task is discarded. Experiments need to examine the indicators: resource utilization, system execution time, task success rate. The standard processing capability for resource providers $Standard_P$ and standard processing capability for resource consumers $Standard_C$ collects the minimum value in the system as the standard, which gives the relative disparity of the processing capability in the system. Through the analysis and experiment, the best empirical parameters are: $\alpha = 0.5$, $\beta = 0.2$, $\gamma = 0.2$, $TL = 90$, $RL = 5$.



(a). short supply (b). supply and demand balance

Fig.7. Comparison of HIIL-CDA and CDA performance

Fig.7(a) shows the performance comparison of CDA and HIIL-CDA. HIIL-CDA is superior to CDA in terms of resource utilization. Rising speed indicates that the resources are fully utilized. After 90 seconds, the descending speed is fast, which indicates that the load is relatively balanced. The end of the resource is similar. In terms of task success rate, HIIL-CDA is much higher than CDA, because the HIIL-CDA price adjustment mechanism can adjust the price of the buyers and sellers to trade within the number *badLimit* (experiment value of 5). CDA has given up the task after five cycles. In the case of insufficient demand, HIIL-CDA is better than CDA. Compared with Fig.7(a), the advantages of the resource utilization are more obvious in Fig.7(b), the gap between the execution time is reduced, and the gap of the success rate between the tasks execution has been greatly reduced. The execution time of HIIL-CDA is slower than CDA, the reason is that HIIL-CDA performs more tasks. The minimum time of the task in the system is 0.12 seconds and the longest is 10 seconds. The average gap between the execution time of CDA and HIIL-CDA is about 7 seconds in the five experiments which includes about 1200 tasks. The execution time is acceptable.

In order to compare HIILDPM and ZI, we design ZI-CDA Which the pricing part adopts ZI[8] and the distribution part adopts CDA matching rule. In order to get ideal price adjustment parameter, a great deal of analysis and experimentation has been done. The experimentation shows that ZI-CDA has better resource utilization and lesser system execution time. In this paper, the ZI strategy adopts the fixed price adjustment method, the resource pricing adjustment parameter β is 0.5, and the task pricing adjustment parameter γ is 0.1. Fig.11 shows that HIILDPM has a higher resource utilization and task execution success rate than the ZI strategy in the task-intensive environment.

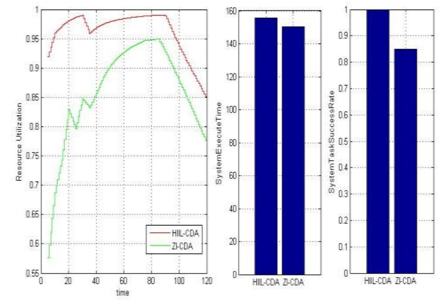


Fig.11. HIIL-CDA and ZI-CDA performance comparison

V. Conclusions

This paper proposes a distributed resource pricing method HIILDPM based on historical transaction records and individual load. The aim is to improve the utilization of resources and match the resources transaction quickly and effectively. The experimental results show that the HIILDPM has higher resource utilization rate and task execution success rate, and the pricing is more reasonable. But there are still some problems need to be studied, such as the selection and the value of the parameters. These will be our future work.

References

[1] Walras L, Dockès P, Walras C A E L. Eléments d'économie politique pure, ou, Théorie de la richesse sociale [M]. Verlag Wirtschaft und Finanzen, 1988, pp.1-20.

[2] Wolski R, Plank J S, Brevik J, Bryan T. Analyzing market-based resource allocation strategies for the computational grid [J]. International Journal of High Performance Computing Applications, vol. 15, no. 3, 2001, pp. 258-281.

[3] Cheng J Q, Wellman M P. The WALRAS algorithm: A convergent distributed implementation of general equilibrium outcomes [J]. Computational Economics, vol. 12, no. 1, 1998, pp. 1-24.

[4] Pourebrahimi B, Ostadzadeh S A, Bertels K. Resource Allocation in Market-Based Grids Using a History-Based Pricing Mechanism [M]. Advances in Computer and Information Sciences and Engineering. Springer Netherlands, 2008, pp. 97-100.

[5] Das R, Hanson J E, Kephart J O, Tesauro G. Agent-human interactions in the continuous double auction [C]. International Joint Conference on Artificial Intelligence. LAWRENCE ERLBAUM ASSOCIATES LTD, vol. 17, no. 1, 2001, pp. 1169-1178.

[6] Rust J, Miller J H, Palmer R. Behavior of trading automata in a computerized double auction market[C]. The Double Auction Market– Institutions, Theories, and Evidence", Proceedings Volume, Santa Fe Institute Studies in Sciences of Complexity. 1993, pp. 1-40.

[7] Vickrey W. Counterspeculation, auctions, and competitive sealed tenders [J]. The Journal of finance, vol. 16, no. 1, 1961, pp. 8-37.

[8] Zhan W J, Zhang J L, Yang J, Wang S Y et al. k-ZI: A General Zero-Intelligence Model in Continuous Double Auction[J]. International Journal of Information Technology & Decision Making, vol. 1, no. 4, 1998, pp. 673-692.

[9] Buyya R, Murshed M. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing [J]. Concurrency & Computation Practice & Experience, vol. 14, no. 13-15, 2002, pp. 1175-1220.