

Cost-aware Grid Workflow Resource Allocation

Pengcheng Xiong and Yushun Fan

Department of Automation, Tsinghua University, Beijing 100084, China
 xpc03@mails.tsinghua.edu.cn and fanyus@tsinghua.edu.cn

Abstract

Smart and swift resource allocation is a fundamental issue to accomplish high performance on grid workflow. In this paper, we study the cost-aware grid workflow resource allocation problem based on a market model of grid resource management architectures. We model the problem as the Multiple Choice Knapsack Problem (MCKP) and design the resource allocation optimization algorithm to minimize the average turnaround time of the grid workflow. The complexity analysis shows that the optimization algorithm leads to more efficient and appropriate resource allocation than many current algorithms.

1. Introduction

Grid workflow [1-2] is a composition of grid application services which execute on heterogeneous and distributed resources in a well-defined order to accomplish a specific goal. The performance of the grid workflow can be evaluated through the ability to meet requirements with respect to some key performance indicators. In this paper, we focus on discussing how to improve the performance of grid workflow through minimizing the average turnaround time.

Grid workflow turnaround time is related to the execution time of all the grid services involved. Execution of a grid service also has cost, which depends on how many resources allocated in order to execute the service. Although more resources probably imply shorter turnaround time for the allocated grid service, in a computational market environment, grid workflow users want to maximize their return-on-investment. This necessitates a grid resource management system that provides appropriate and fast algorithms to minimize the grid workflow turnaround time and yet meets the computational cost that users agree to pay.

Currently, there are three alternative models [3], i.e., hierarchical, abstract owner, and market model for grid resource management architectures. From

economic standpoint, Buyya *et al.* [3] develop Grid Economy as a combination of Globus and GRACE services. From workflow standpoint, Li *et al.* [4] propose a computation algorithm for the lower bound of average turnaround time of workflow services through resource availability and workload analysis. However, few of the above research have made substantial efforts in combining the internal execution process logic of grid workflow service nodes with an efficient enough resource allocation algorithm to catch up with the highly varying grid environment. In this paper, we take both the grid workflow turnaround time and the limited grid resource budget into consideration. Moreover, the algorithm we propose to solve this kind of problem is also proved to be especially efficient and suitable.

2. Model description

2.1. Extended market model

Definition 1. An extended market model (EMM) is a five tuple $(G, AO, R, F_{go}, F_{or})$, where:

- 1) $G = \{a_1, a_2, \dots, a_u\}$ is a set of grid workflow services, where a_i is a grid workflow service defined in grid workflow perspective;
- 2) $AO = \{o_1, o_2, \dots, o_v\}$ is a set of abstract owners, where o_i is an abstract owner defined in grid resource broker perspective;
- 3) $R = \{r_1, r_2, \dots, r_q\}$ is a set of resource pools, where each r_i denotes a resource pool defined in the resource perspective;
- 4) $F_{go} \subseteq G \times AO$ denotes the mapping relation between grid workflow perspective and grid resource broker perspective;
- 5) $F_{or} \subseteq O \times R$ denotes the mapping relation between grid resource broker perspective and resource perspective.

The extended market model is shown in Fig. 1. In the first layer, directed graph is adopted to specify the process control structure of a grid workflow model. The second layer is the resource broker perspective, in which each grid workflow service is appointed a kind of abstract owners for its executing. The grid middleware and domain resource manager are contained in the resource perspective of the third layer. In this layer, a class of individual resource agents (e.g., high performance computers) that have the same skills and capability are grouped to form a resource pool.

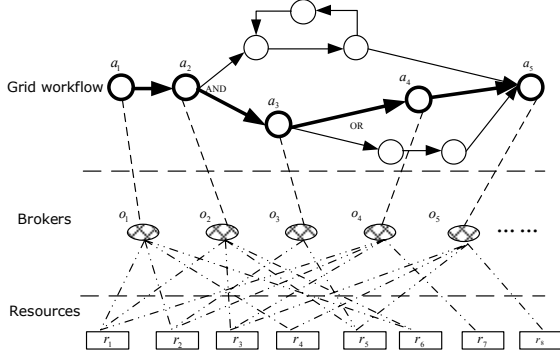


Fig. 1. Graph representation of an example EMM

The dotted lines describe the mapping relations among these three layers. Suppose that the execution of a grid workflow service needs the support of a kind of abstract owners, e.g., suppose that the execution of grid workflow service a is appointed to abstract owner o . Abstract owner o is supported by n_1 individual resource agents of r_1 , n_2 of r_2 , ... and n_k of r_k , which can be denoted as $((r_1, n_1), (r_2, n_2), \dots, (r_k, n_k))$. If the cost for an individual resource agent in resource pool r_i is c_i , then the total cost for single abstract owner o

is computed to be $\sum_{i=1}^k n_i \cdot c_i$.

2.2. Critical path

The grid workflow layer of an EMM is composed of grid workflow services interconnected by various structures [4]. A workflow instance presents an actual process in execution. Suppose that each grid workflow service has exponential execution time, the arrival process of user's service requests is a Poisson process and the queue discipline is first come-first served. We can model a grid workflow as an M/M/c queuing network. The critical path is a sequence of grid workflow services from the beginning to the end of a grid workflow that has the longest average execution time. Grid workflow services on the critical path are called critical services. We directly adopt the innermost control structure first (ICSF) method proposed by Son [5] to find the critical path of a grid

workflow. Figure 2 shows an example critical path of a workflow which is made up of a_{1-n} , where λ_i , u_i and z_i denotes the instance arrival rate, single abstract owner service rate and minimum number of abstract owners for a_i , respectively. According to queuing theory, we have $z_i = \text{Roundup}(\lambda_i / u_i)$ where the function $\text{Roundup}(\lambda_i / u_i)$ returns the nearest integer no less than λ_i / u_i .

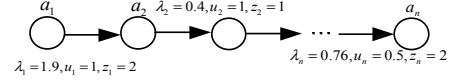


Fig. 2. Graph representation of an example critical path of a workflow

2.3. Problem model

According to queuing theory, the average execution time for a critical service can be reduced by increasing the number of abstract owners while the number of abstract owners is constraint to the limited cost. Suppose the available budget as C , the problem is how to minimize the average execution time for the critical path of the grid workflow under the economic constraint. This problem can be modeled as a Multiple Choice Knapsack Problem (MCKP).

For a critical path composed of n services, we model the resource allocation problem as MCKP in the following way. We treat the kinds of abstract owners as the classes in MCKP. Besides the minimum number of abstract owners, each number of extra abstract owners is treated as a candidate for allocation. Thus, every candidate represents an item in that class. Since abstract owners are supported by a set of resource agents through F_{or} , the total cost of resource agents is the weight in MCKP. Zero minus the execution time of a specific service represents the profit of the items in MCKP. The objective is to minimize the turnaround time of the critical path under the total cost constraint C . The problem can be presented formally as:

$$\text{Max} \sum_{i=1}^n \sum_{j \in S_i} ((-T_{ij}) \cdot x_{ij}) \quad (1)$$

$$\text{s.t.} \sum_{i=1}^n \sum_{j \in S_i} (c_{ij} \cdot x_{ij}) \leq C \quad (2)$$

$$\sum x_{ij} = 1, i = 1, \dots, n, j \in S_i \quad (3)$$

$$x_{ij} = \{0, 1\}, i = 1, \dots, n, j \in S_i \quad (4)$$

T_{ij} represents the average execution time for workflow service a_i supported by $z_i + j$ abstract owners. T_{ij} can be computed as :

$$T_{ij} = \frac{1}{u_i} + \frac{((z_i + j) \cdot \rho)^{z_i + j}}{(z_i + j) \cdot u_i \cdot (1 - \rho)^2 \cdot (z_i + j)! \left(\sum_{k=0}^{z_i + j - 1} \frac{((z_i + j) \cdot \rho)^k}{k!} + \frac{((z_i + j) \cdot \rho)^{z_i + j}}{(z_i + j)! (1 - \rho)} \right)}$$

, where $\rho = \lambda_i / (u_i \cdot (z_i + j))$

c_{ij} denotes the cost for workflow service a_i with $z_i + j$ abstract owners.

C represents the total available budget.

3. Pisinger's algorithms

Pisinger proposes a minimal algorithm for MCKP in [6]. The first step is to delete the LP-dominated solutions. The second step is to relax the integrity constraint (4) to $0 \leq x_{ij} \leq 1$ and therefore transform the MCKP problem to a Linear Multiple Choice Knapsack Problem (LMCKP). We solve this LMCKP and can obtain the optimal solution b_i in each class S_i . If for each b_i , we have $x_{ib_i} = 1$, then the optimal solution for this LMCKP is also the optimal solution for MCKP. Otherwise, there exists and only exists a class S_a containing two fractional variables $x_{ab_{a_1}}$ and $x_{ab_{a_2}}$ satisfying $x_{ab_{a_1}} + x_{ab_{a_2}} = 1$. Under this condition, the third step takes the class S_a as the initial core and treats b_i as the initial solution set. Then we define the positive and negative gradient λ_i^+ and λ_i^- for each class S_i , $i \neq a$ as follows:

$$\lambda_i^+ = \text{Max}_{j \in S_i, c_{ij} > c_{ib_i}} \frac{T_{ib_i} - T_{ij}}{c_{ij} - c_{ib_i}}, i = 1, 2, \dots, k, i \neq a$$

$$\lambda_i^- = \text{Max}_{j \in S_i, c_{ij} < c_{ib_i}} \frac{T_{ij} - T_{ib_i}}{c_{ib_i} - c_{ij}}, i = 1, 2, \dots, k, i \neq a$$

We sort the set $L^+ = \{\lambda_i^+\}$ and $L^- = \{\lambda_i^-\}$ in decreasing and increasing order respectively. Then we expand the core by alternatively including a new class S_i through selecting λ_i^+ with the largest value in L^+ or λ_i^- with the smallest value in L^- , beginning from the initial core. According to the upper bound, we can trim off unfeasible class prior to adding a new class to the core. Therefore, we derive a core composed of a set

of classes, i.e., $\text{Core} = \{S_{v_1}, S_{v_2}, \dots, S_{v_m}\}$. The corresponding partial vector can be formulated as $Y_{\text{Core}} = \{(y_1, y_2, \dots, y_m) \mid y_i \in \{0, 1, 2, \dots, M_{v_m}\}, i = 1, 2, \dots, m\}$, where y_i represents the choice of class S_i , satisfying $x_{iy_i} = 1$ and $\forall j \neq y_i, x_{ij} = 0$. We mark the state as a three tuple (μ_i, π_i, δ_i) , where δ_i is a representation of Y_{Core} , μ_i and π_i are given below. The optimal solution for MCKP is obtained after all the classes have been selected.

$$u_i = \sum_{S_i \in \text{Core}} c_{iy_i} + \sum_{S_i \notin \text{Core}} c_{ib_i}$$

$$\pi_i = \sum_{S_i \in \text{Core}} T_{iy_i} + \sum_{S_i \notin \text{Core}} T_{ib_i}$$

4. Case study

Suppose that the critical path of the workflow example is composed of five individual services connected by a bold line in Fig. 1, the budget constraint $C = 200$. The arrival rate and single abstract owner service rate for every service is shown in Tab. 1. Individual resource agent and price are presented in Tab. 2. Mapping relationship F_{or} is in Tab. 3.

Table 1. Arrival rate λ_i , single abstract owner service rate u_i and minimal number of abstract owners z_i

Services on the critical path	Arrival rate λ_i	Single abstract owner service rate u_i	Minimal number of abstract owners z_i
a_1	2.00	1.10	2
a_2	0.55	1.00	1
a_3	1.43	0.80	2
a_4	1.97	1.15	2
a_5	0.76	0.50	2

Table 2. Individual resource agent and price

Resource agent	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
Price	1.20	2.00	1.50	0.80	4.00	5.00	3.00	0.5

Table 3. Mapping relationship F_{or}

Abstract owners	Individual resource agent	Total cost for abstract owners
o_1	$((r_1, 2), (r_2, 1), (r_4, 3), (r_6, 1))$	11.8
o_2	$((r_1, 1), (r_3, 2), (r_5, 1), (r_6, 2))$	18.2
o_3	$((r_2, 2), (r_3, 2))$	12
o_4	$((r_1, 1), (r_2, 4), (r_3, 1), (r_7, 2))$	16.7
o_5	$((r_3, 2), (r_4, 1), (r_5, 2), (r_6, 6))$	14.8

Table 4. Total cost of abstract owners and the execution time for grid service $c_{ij} [T_{ij}]$

Grid Service	Add 0 abstract owner	Add 1 abstract owner	Add 2 abstract owners	Add 3 abstract owners	Add 4 abstract owners	Add 5 abstract owners	Add 6 abstract owners
a_1	23.6 [5.2381]	35.4 [1.1880]	47.2 [0.9642]	59.0 [0.9211]	70.8 [0.9116]	82.6 [0.9096]	94.4 [0.9092]
a_2	18.2 [2.2222]	36.4 [1.0818]	54.6 [1.0080]	72.8 [1.0007]			
a_3	24 [6.2124]	36 [1.6102]	48 [1.3212]	60 [1.2654]	72 [1.2332]	84 [1.2506]	
a_4	33.4 [3.2645]	50.1 [1.0847]	66.8 [0.9118]	83.5 [0.8785]	100.2 [0.8714]		
a_5	29.6 [4.7348]	44.4 [2.3298]	59.2 [2.0626]	74.0 [2.0122]	88.8 [2.0022]		

The initial core contains $\{18.2 [2.2222], 36.4 [1.0818], 54.6 [1.0080], 72.8 [1.0007]\}$. We compute the positive and negative gradient λ_i^+ and λ_i^- for S_i , $i \neq 2$. We then sort the sets $L^+ = \{\lambda_i^+\}$ and $L^- = \{\lambda_i^-\}$ in decreasing and increasing values respectively in the following Tab. 5 and Tab. 6.

Table 5. L^+

No.	Service/Class	$d(-T)/dc$
1	a_3 / S_3	0.02408
2	a_1 / S_1	0.01896
3	a_5 / S_5	0.01805
4	a_4 / S_4	0.01035

Table 6. L^-

No.	Service/Class	$d(-T)/dc$
1	a_4 / S_4	0.1305
2	a_5 / S_5	0.1625
3	a_1 / S_1	0.3432
4	a_3 / S_3	0.3835

The set of partial vector Y_{Core} in the initial core has 4 states, i.e., $Y_{Core} = \{(\mu_i, \pi_i, \delta_i)\} = \{(184.1, 8.4349, 1), (202.3, 7.2945, 2), (220.5, 7.2207, 3), (238.7, 7.2134, 4)\}$. After doing state reduction by the upper bound test, we obtain $Y_{Core} = \{(184.1, 8.4349, 1)\}$, which denotes that the shortest average turnaround time is 8.4349. The final result is to allocate 3, 1, 4, 3, 3 abstract owners to a_{1-5} respectively, and the shortest average turnaround time is 8.1459.

5. Conclusion

In this paper, we extend the market model towards grid workflow turnaround time quantitative analysis and optimization. We model the problem as MCKP and adopt efficient Pisinger's algorithm. Actually, the problem's convexity characteristic provides additional convenience of using the algorithm. Moreover, the proposed concepts and algorithms can be readily put into industrial applications.

Acknowledgements

This paper is supported by the National High Technology Research and Development (863) Program of China under Grant 2006AA04Z151 and the China National Science Foundation under Grant 60674080.

References

- [1] H. Zhuge, T. Cheung, and H. Pung, "A timed workflow process model", *Journal of Systems and Software*, Vol. 55, Iss. 3, 2001, pp. 231-243.
- [2] H. Zhuge, "China's E-Science Knowledge Grid Environment", *IEEE Intelligent Systems*, Vol. 19, Iss. 1, 2004, pp. 13-17.
- [3] R. Buyyat, S. Chapin and D. DiNucci, "Architectural Models for Resource Management in the Grid", *Proc. of the First IEEE/ACM International Workshop on Grid Computing*, Bangalore, India, 2000, pp. 1-13.
- [4] J. Li, Y. Fan and M. Zhou, "Performance modeling and analysis of workflow", *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol. 34, Iss. 2, Mar. 2004, pp. 229-242.
- [5] J. H. Son and M. H. Kim, "Improving the performance of time-constrained workflow processing", *Journal of Systems and Software*, Vol. 53, 2001, pp. 211-219.
- [6] D. Pisinger, "A minimal algorithm for the Multiple-choice Knapsack Problem", *European Journal of Operational Research*, 83, 1995, pp. 394-410.