Towards Scheduling Data-Intensive and Privacy-Aware Workflows in Clouds

Yiping Wen[1,2(✉)], Wanchun Dou[1], Buqing Cao[2], and Congyang Chen[2]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
ypwen81@gmail.com, douwc@nju.edu.cn
[2] Key Laboratory of Knowledge Processing and Networked Manufacture, Hunan University of Science and Technology, Xiangtan, China

Abstract. Nowadays, business or scientific workflows with a massive of data are springing up in clouds. To avoid security and privacy leakage issues, users' privacy or sensitive data may be restricted to being processed in some specified and trusted cloud datacenters. Meanwhile, users may also pay attention to the cost incurred by renting cloud resources. Therefore, new workflow scheduling algorithms should be developed to achieve a balance between economically utilizing the cloud resources and protection of users' data privacy and security. In this paper, we propose a cost-aware scheduling algorithm for executing multiple data-intensive and privacy-aware workflow instances in clouds. Our proposed algorithm is based on the strategy of batch processing, the ideas of simulated annealing algorithm and the particle swarm optimization, the coding strategy of which is devised to minimize the total execution cost while meeting specified privacy protection constraints. The experimental results demonstrate the effectiveness of our algorithm.

Keywords: Privacy protection · Cloud · Workflow scheduling · Cost · Batch processing · Particle swarm optimization

1 Introduction

Workflow scheduling in clouds is an important research topic [1], which tries to map the workflow tasks to the dynamically provisioned resources based on different functional and non-functional requirements. However, existing cloud workflow scheduling algorithms normally do not consider the security requirement of meeting privacy protection constraints when making resource allocation decisions, yet protection of privacy or sensitive data involved in tasks is important in the cloud workflows for business or scientific purpose, which may contain one or several data intensive tasks (with huge data) in the big data era. These data intensive cloud workflows usually are abstract of cross-organizational business processes which include trade secrets or personal privacy data. For security reason, they may not be allowed to be scheduled to some cloud datacenters though these cloud datacenters can speed up more efficiently or much cheaper. For example, financial data and customers consuming data may be secret information for some enterprises and they usually may be restricted to being processed in specified

S. Wang and A. Zhou (Eds.): CollaborateCom 2016, LNICST 201, pp. 474–479, 2017.
DOI: 10.1007/978-3-319-59288-6_44

and trusted datacenters to provide privacy protection. Hence, scheduling algorithms for such privacy-aware workflow in clouds should be developed to solve such new issue.

Based on our previous works in [2], this paper proposes a cost-aware scheduling algorithm for executing multiple data-intensive and privacy-aware workflow instances in clouds, which is called BCP-PSO and aims at optimizing the total workflow execution cost while meeting specified privacy protection constraints.

To the best of our knowledge, our work is the first approach considering both privacy protection constraints and batch processing strategy in workflow scheduling. It can be viewed as an improvement to our previous work in [2], which only focuses on optimizing the execution cost of one workflow instance with privacy protection constraints while we focus on that of a set of concurrent workflow instances in this paper. In addition, we further employ the particle swarm optimization (PSO) [3], the ideas of simulated annealing (SA) algorithm [4] and batch processing strategy to reduce the execution cost. The resource utilization is also improved by further utilize the idle time fragments in the rented virtual machine instances according to the time unit-based pricing model.

2 Design of Scheduling Algorithm

In our work, we introduce the ideas of the SA into PSO and construct a variant inertia weight function featured in annealing mechanism. Meanwhile, we adopt the batch processing strategy to optimize scheduling a group of instances of the same task simultaneously. The pseudo code for our proposed algorithm is described as Fig. 1, and the following sections provide their related steps in detail.

2.1 Group Unscheduled Ready Task Instances

The step of grouping unscheduled ready task instances (Line 3) adopts the strategy of batch processing to handle task instances from multiple concurrent instances with the same workflow model. A ready task instance is either the instance of the entry task in the workflow or the other task's instance whose predecessors have all been allocated. Each ready task instance will belong to only one group and all task instances in the same group have the same task type. By grouping task instances, cloud resources can be reused to reduce execution cost, which will be explained in Sect. 2.3.

2.2 Coding and Privacy Protection Constraints Handling Strategy

In PSO, each alternative solution is called as a particle. Therefore, we need to establish the meaning of a particle and deal with the privacy protection constraints to get alternative feasible solutions for our problem.

To promote efficiency, we firstly create a feasible resource list according to privacy protection constraints and the corresponding task type of instances in current task instance group. For example, if all task instances in a group are instances of the task t_k with privacy or sensitive data, which are restricted to being processed only in the datacenter dc_i, only the cloud resources in dc_i are included in the feasible resource list and

Algorithm 1. Pseudo code of BCP-PSO algorithm

Input: Set of resources, set of workflow instances, *N*: number of particles
Output: *FS*: a workflow scheduling solution
1: $FS \leftarrow \varnothing$, set initial parameters for PSO and SA;
2: **While** there are unscheduled "ready" task instances
3: Generate unscheduled "ready" task instance group set *RTIG* so that instances of the same task are in the same group;
4: **For** each group in *RTIG* **do**
5: Generate a new feasible resource list *RL* according to privacy protection constraints and the task type of current group;
6: Set particle dimension equal to the size of current group;
7: Initialize particles position randomly from {1, …, |*RL*|} and velocity randomly; // |*RL*| is the number of resources in *RL*
8: **Repeat**
9: **For** each particle i = 1 to N **do**
10: Calculate its fitness value;
11: If current fitness value is better than the fitness value of its $pbest_i$, set current location as the new $pbest_i$;
12: Modify the inertia weight and update the velocity and position of each particle;
13: **End for**
14: Modify *gbest* by the particle with the best fitness value of all the particles, and annealing temperature $T = \alpha \cdot T$;
15: **Until** maximum iteration is satisfied
16: Add the schedule of task instances in current group to *FS*;
17: **End for**
18: **End while**
19: **return** *FS*

Fig. 1. Pseudo code of BCP-PSO algorithm

each cloud resource is assigned a unique positive integer index for allocating them to task instances in such group.

Our coding strategy is devised based on the generated feasible resource list and task instance group. We set the dimension of each particle equal to the size of task instance group and each position in each particle represents a task instance, the value of which represents the index of a cloud resource in feasible resource list.

2.3 Generate a Schedule of a Task Instance Group

According to the coding strategy described above, we can convert a particle's position into a schedule of a task instance group and calculate the total execution cost so far. Because commercial cloud providers typically charge users by an hourly-based pricing model, it may leave much idle time. For example, suppose t_{ik} and t_{jk} are two instances of the task t_k; if the processing time of is 20 min, we still lease the cloud resource (e.g., VM) for one hour. Thus, the VM will be in idle for 40 min which can be reused by t_{jk}. In this case, if t_{ik} and t_{jk} have been mapped to the same VM, this VM can be reused

to reduce execution cost. By using PSO and the strategy of batch processing, VM reuse can be accomplished more effectively for scheduling multiple workflow instances in clouds.

2.4 Update Velocity and Position of Particles

In the iterative phase (Line 8–15) of our algorithm, to ensure that the search is done inside the positive integer space, the velocity and position of a particle are updated based on the Eqs. 1 and 2 respectively:

$$v_i^{k+1} = \left[\omega_i^k \cdot v_i^k + c_1 \cdot r_1 \cdot \left(pbest_i^k - x_i^k\right) + c_2 \cdot r_2 \cdot \left(gbest^k - x_i^k\right)\right] \tag{1}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{2}$$

where v_i^{k+1} is the velocity of particle i in iterative $k+1$, ω_i^k is the inertia weight of particle i in iterative k, c_1 and c_2 are two positive numbers termed learning factors, r_1 and r_2 are two random numbers with uniform distributed in the range [0, 1], x_i^k is the position of particle i in iterative k, $pbest_i^k$ is the individual best position for particle i after k iterations, $gbest^k$ is the best position for all the particles after k iterations.

The inertia weight ω_i^k in Eq. 1 keeps particle i with the movement inertia. When ω_i^k is larger, particle i has better ability to search for a global optimum solution, otherwise the local search capability of particle i is better. Therefore, we dynamically adjust ω_i^k on the basis of the ideas of the SA to improve the probability and particle's ability of finding the global or near optimum solution. ω_i^k is updated based on Eqs. 3 and 4:

$$\omega_i^k = \begin{cases} 1 + \dfrac{ran}{2} & \rho_i^k \geq ran \\ \dfrac{ran}{2} & \rho_i^k < ran \end{cases} \tag{3}$$

$$\rho_i^k = \begin{cases} 1 & fitness\left(x_i^{k-1}\right) > fitness\left(x_i^k\right) \\ e^{\dfrac{fitness\left(x_i^{k-1}\right) - fitness\left(x_i^k\right)}{T}} & fitness\left(x_i^{k-1}\right) \leq fitness\left(x_i^k\right) \end{cases} \tag{4}$$

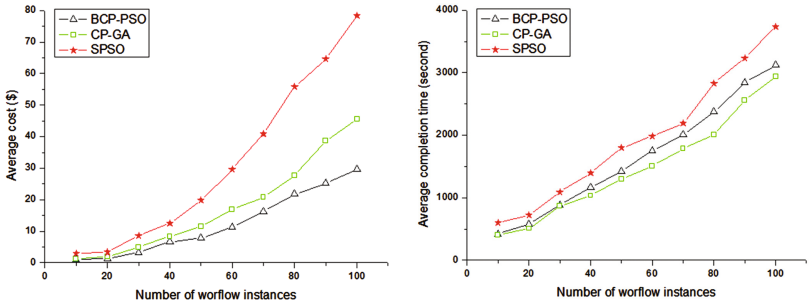
where ran is a random number with uniform distributed in the range [0, 1], T represents current annealing temperature. If $fitness\left(x_i^{k-1}\right) > fitness\left(x_i^k\right)$, meaning that, the position of particle i in iterative k is better than the previous iterative $k-1$ for the fitness function.

Besides, the updates of velocity and position are liable to cause particles to exceed the search boundaries. Our algorithm adopts the handling method in [5] to keep particles within the search space.

3 Experiments and Evaluation

In order to test the algorithm performance, we use the CloudSim framework to simulate a cloud environment and make up three datacenters and ten virtual machines with four types. The CP-GA algorithm [2] and SPSO algorithm are tested against the proposed BCP-PSO algorithm, where the SPSO algorithm is based on PSO [3] while using the coding and privacy protection constraints handling strategy of the BCP-PSO algorithm to schedule multiple data-intensive and privacy-aware workflow instances in clouds. In our experiments, the numbers of particles are equal to 20, and the values of learning factors are equal to 1.49445.

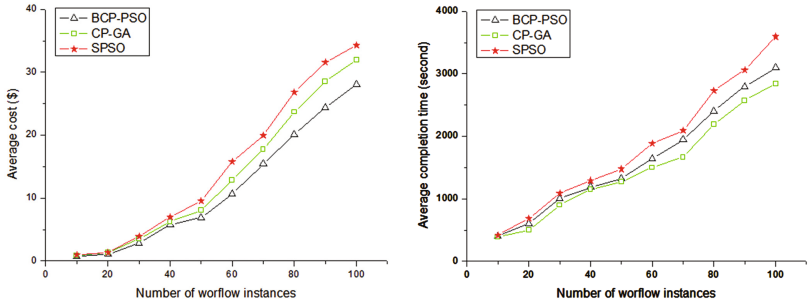
Figure 2 demonstrates the evaluation results of three algorithms with specified privacy protection constraints on task t_3 and task t_5. Figure 2a shows that the BCP-PSO algorithm outperforms the CP-GA and SPSO algorithms in terms of the cloud resources cost of workflow instances. With the growth of the size of workflow instances, the optimization on cost of BCP-PSO is better. The main reason is that the BCP-PSO algorithm adopts the batch processing strategy to reuse the VM and reduce execution cost. However, this will increase the completion time of workflow instances compared to the CP-GA algorithm, which is shown as Fig. 2b.



(a) Average cloud resources cost (b) Average completion time

Fig. 2. Evaluation results of three algorithms with specified privacy protection constraints

Figure 3 demonstrates the evaluation results of these three algorithms without specified privacy protection constraints on task t_3 and task t_5. In terms of the cloud resources cost of workflow instances, the BCP-PSO algorithm also outperforms the other two algorithms.



(a) Average cloud resources cost (b) Average completion time

Fig. 3. Evaluation results of three algorithms without specified privacy protection constraints

4 Conclusion

In this paper, we analyze the cost optimization problem of scheduling workflow with privacy protection constraints and propose a cost-aware scheduling algorithm for executing multiple data-intensive and privacy-aware workflow instances in clouds. In our algorithm, we use the strategy of batch processing to group task instances according to their task type, and incorporate the privacy protection constraints to devise the coding strategy of particles. We also introduce the ideas of the SA into PSO and construct a variant inertia weight function to overcome premature convergence. The comparative experiments show the effectiveness of our algorithm.

Acknowledgments. This paper was supported by National Natural Science Fund of China, under grant number 61402167, 61572187, 61402168, and National Science and Technology Support Project of China, under grant number 2015BAF32B01.

References

1. Smanchat, S., Viriyapant, K.: Taxonomies of workflow scheduling problem and techniques in the cloud. Future Gener. Comput. Syst. **52**, 1–12 (2015)
2. Chen, C., Liu, J., Wen, Y., Chen, J., Zhou, D.: A hybrid genetic algorithm for privacy and cost aware scheduling of data intensive workflow in cloud. In: Wang, G., Zomaya, A., Perez, G.M., Li, K. (eds.) ICA3PP 2015. LNCS, vol. 9528, pp. 578–591. Springer, Cham (2015). doi: 10.1007/978-3-319-27119-4_40
3. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neutral Networks. pp. 1942–1948. IEEE Service Center, Piscataway (1995)
4. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
5. Li, Z.J., Ge, J.D., Yang, H.J., Huang, L.G., Hu, H.Y., Hu, H., Luo, B.: A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds. Future Gener. Comput. Syst. **65**, 140–152 (2016)