



# BTS: Balanced Task Scheduling Strategy Based on Multi-resource Prediction and Allocation in Cloud Environment

Yongzhong Sun<sup>1</sup>, Kejiang Ye<sup>1</sup>(✉), Wenbo Wang<sup>2</sup>, and Cheng-Zhong Xu<sup>3</sup>

<sup>1</sup> Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences,  
Shenzhen 518055, China

{yz.sun1,kj.ye}@siat.ac.cn

<sup>2</sup> Khoury College of Computer Sciences, Northeastern University,  
Seattle, WA 98109, USA

wang.wenbo@husky.neu.edu

<sup>3</sup> Faculty of Science and Technology, University of Macau, Macau, China  
czxu@um.edu.mo

**Abstract.** Cloud computing is a new computing paradigm equipped with large-scale servers to satisfy diverse application demands. Managing and scheduling various application tasks on cloud servers is very challenging. In this paper, we propose a Balanced Task Scheduling (BTS) strategy by combining multi-objective particle swarm optimization and time series prediction model to achieve a better load balance among cloud servers. We not only consider the current server load which is used by most existing scheduling methods, but also take the future load change prediction into account. Experiments on the public Alibaba cluster trace with 1310 servers show that the proposed strategy can achieve a more balanced resource utilization.

**Keywords:** Load balancing · Workload prediction · Task scheduling

## 1 Introduction

Despite the adoption of various resource management systems that use typical scheduling algorithms based on instantaneous resource availability during the scheduling, the ability to reliably distribute application tasks among cloud servers remains deficient. According to the analysis of Alibaba cluster data [3], cloud servers have a significant spatial imbalance and time imbalance. Due to the limits of existing task scheduling methods, this paper proposes a balanced task scheduling strategy based on multi-resource prediction and allocation to achieve a better load balance among cloud servers.

The main contributions of this paper are: (i) According to the load feedback sampled periodically, we forecast the future load of servers through a time series prediction model - Prophet [7]. Then we use a multi-objective particle swarm

optimization algorithm - OMOPSO [8] to determine the mapping relationship between the tasks and the servers from the predicted load, actual load, and load threshold. (ii) We use the Alibaba cluster trace with 1310 servers as the test dataset to evaluate the prediction accuracy and also perform the load balance analysis to verify the effectiveness of the task scheduling strategy. Experimental results show that the proposed strategy can achieve a more balanced CPU and memory utilization.

## 2 Problem Description

**Definition 1.** Server and its resource utilization vector. The data center has  $n$  servers  $S_i, i \in [1, n]$ . Vector  $\vec{S}_i^{cur} = (S_{i,CPU}^{cur}, S_{i,Mem}^{cur})$  represents the current resource utilization of different servers in the data center,  $S_{i,CPU}^{cur}$  is the current CPU utilization of server  $S_i$ ,  $S_{i,Mem}^{cur}$  is the current memory utilization of server  $S_i$ . Vector  $\vec{S}_i^{next} = (S_{i,CPU}^{next}, S_{i,Mem}^{next})$  represents the predicted resource utilization of different servers in the data center at the next time.

**Definition 2.** Batch task and its resource occupancy rate. The number of batch tasks that need to be deployed to the server at a given time is  $m$ ,  $B_j, j \in [1, m]$  represents a batch task,  $B_{j,CPU}$  is the CPU requirement of  $B_j$ ,  $B_{j,Mem}$  is the memory requirement of  $B_j$ .

**Definition 3.** Batch tasks to servers deployment matrix. The deployment relationship between the batch tasks and servers can be expressed as a matrix  $E = (e_{ij})_{n \times m}$ . When batch task  $B_j$  is deployed to server  $S_i$ ,  $e_{ij} = 1$ , otherwise  $e_{ij} = 0$ .

**Definition 4.** Server and its current utilization estimate. For server  $S_i$ , its current CPU utilization estimate is the sum of  $S_{i,CPU}^{cur}$  and the CPU resource requested for all batch tasks deployed on it:  $EST_{i,CPU}^{cur} = S_{i,CPU}^{cur} + \sum_{j=1}^m e_{ij}B_{j,CPU}$ . In the same way, its current memory utilization estimate is  $EST_{i,Mem}^{cur} = S_{i,Mem}^{cur} + \sum_{j=1}^m e_{ij}B_{j,Mem}$ .

**Definition 5.** Server and its next-period utilization estimate. Assume that the batch tasks currently deployed are not finished in the next period. For server  $S_i$ , its next-period CPU utilization estimate  $EST_{i,CPU}^{next}$  is the sum of  $S_{i,CPU}^{next}$  and the CPU resource requested for all the batch tasks currently deployed on it:  $EST_{i,CPU}^{next} = S_{i,CPU}^{next} + \sum_{j=1}^m e_{ij}B_{j,CPU}$ . Its next-period memory utilization estimate  $EST_{i,Mem}^{next} = S_{i,Mem}^{next} + \sum_{j=1}^m e_{ij}B_{j,Mem}$ .

**Problem Model.** By introducing the above definitions, the server load balancing problem can be modeled as a multi-objective optimization problem, whose objective functions:

$$\min(K_{Res}^{cur}) = \min \left( \sqrt{\frac{1}{n} \sum_{i=1}^n \left( EST_{i,Res}^{cur} - \frac{1}{n} \sum_{i=1}^n EST_{i,Res}^{cur} \right)^2} \right), \quad (1)$$

$Res \in \{CPU, Mem\}$

$K_{Res}^{cur}$  is the standard deviation of the current resource utilization estimate for servers of the data center.

The constraint functions are as follows:

$$\sum_{i=1}^n e_{ij} = 1, j = 1, 2, \dots, m \quad (2)$$

indicating that each batch task can only be deployed on one server.

$$EST_{i,Res}^{cur} = S_{i,Res}^{cur} + \sum_{j=1}^m e_{ij} B_{j,Res} < T_{i,Res} \quad (3)$$

$$EST_{i,Res}^{next} = S_{i,Res}^{next} + \sum_{j=1}^m e_{ij} B_{j,Res} < T_{i,Res} \quad (4)$$

represent that when the batch tasks are deployed on the servers, the current and next-period resource utilization cannot exceed the server resource threshold. The resource threshold of server  $S_i$  is  $T_{i,Res}$ .

### 3 Experimental Evaluation

The cluster data released by Alibaba in 2017 is used as the experimental data. It contains 12-h trace information of 1,310 machines, including machine resource usage and batch task workload.

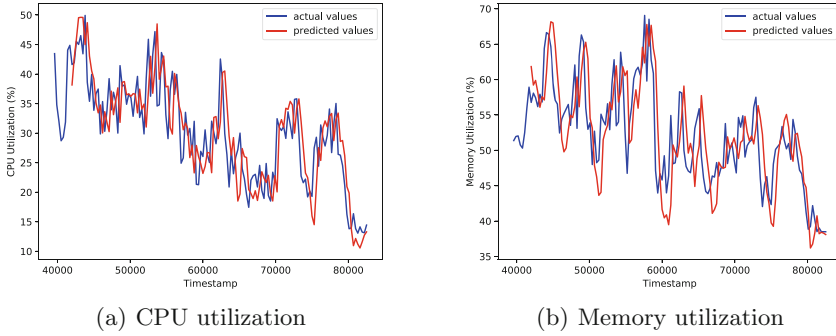
We use the logistic regression model of Prophet for prediction. The model parameters are as follows: capacity is 100%, changepoint\_range is 100%, changepoint\_prior\_scale is 0.2, and n\_changepoint is automatically set by the model. The sliding window mechanism was applied to predict the workload and the length of the window is set to 8.

We first verify the prediction accuracy of the proposed method. Figure 1 shows the actual load and predicted load of a server (id = 600) in the sampling period. The figure shows that the prediction can fit the fluctuation of the machine load very well.

Then, we evaluate the effectiveness of balanced scheduling strategy. We select 4 load sampling time periods from Alibaba cluster data, using the first 5,000 batch tasks in all servers for rescheduling in each time period.

We find the solution to problem (1) by the OMOPSO algorithm under constraints (2)(3)(4). By tracking 4 load sampling timestamps, we get the actual resource utilization  $S_{i,CPU}^{cur}$  and  $S_{i,Mem}^{cur}$  of the machines, and we get the predicted value  $S_{i,CPU}^{next}$  and  $S_{i,Mem}^{next}$  of future resource utilization through the Prophet model. The resource utilization threshold  $T_{i,CPU}$  and  $T_{i,Mem}$  of server  $S_i$  are set to 70% and 90% respectively. The parameters for particle swarm optimization are set as follows:  $w = rand(0.1, 0.5)$ ,  $c_1, c_2 = rand(1.5, 2.0)$ ,  $r_1, r_2 = (0.0, 1.0)$ ,  $polupationSize = 50$  and  $maxEvolution = 1000$ .

The load balancing effect is tested by calculating the standard deviation of the load of cloud servers, and the results are shown in Table 1, where  $K_{CPU}^{orig}$  and



**Fig. 1.** Actual and predicted load comparison of machine id 600

$K_{Mem}^{orig}$  represent the standard deviation of the CPU load and memory load of the machines when the original scheduling strategy is adopted. In the case of using the proposed scheduling strategy, the load balance of each experimental group is improved compared with the original scheduling strategy.

**Table 1.** Load balancing effect of two scheduling strategies

Group	$K_{CPU}^{cur}$	$K_{CPU}^{orig}$	$\Delta K_{CPU}$	$K_{Mem}^{cur}$	$K_{Mem}^{orig}$	$\Delta K_{Mem}$
A	6.899	7.255	<b>-0.356</b>	10.376	11.179	<b>-0.803</b>
B	6.409	6.816	<b>-0.407</b>	11.715	12.351	<b>-0.636</b>
C	7.140	7.969	<b>-0.829</b>	10.726	11.195	<b>-0.469</b>
D	7.167	7.520	<b>-0.353</b>	12.265	12.884	<b>-0.619</b>

## 4 Related Work

The intelligent algorithms such as simulated annealing algorithm [9], genetic algorithm [6] and particle swarm optimization [4] are powerful in solving the task scheduling problem under multi-resource constraints. LD et al. [1] propose a dynamic load balancing algorithm HBB-LB based on bees’ foraging behavior, aiming to achieve load balancing across VMs to maximize throughput. The priority of the task in the waiting sequence in the node is considered to minimize the waiting time of the task in the queue. Li et al. [2] propose a cloud task scheduling policy based on Load Balancing Ant Colony Optimization (LBACO) algorithm. The algorithm selects the best resource to perform a task based on the resource state and the size of a given task in the cloud environment. It balances the overall system and minimizes the completion time for a given set of tasks. Ramezani et al. [5] propose a Task-based System Load Balancing method using Particle Swarm Optimization (TBSLB-PSO) that achieves system load

balancing by only transferring extra tasks from an overloaded VM instead of migrating the entire overloaded VM. It significantly reduces the time taken for the load balancing process.

## 5 Conclusion

In order to solve the load balancing problem, this paper proposes a task scheduling strategy based on the combination of multi-objective particle swarm optimization and time series prediction model. The goal of this strategy is to improve load balancing among the cloud servers, and the impact of the current and future load of the servers on task scheduling is also considered. The experiments based on Alibaba cluster trace with 1310 servers show that this scheduling strategy can effectively achieve the goal of reasonable task allocation with a more balanced resource utilization.

**Acknowledgment.** This work is supported by the National Key R&D Program of China (No. 2018YFB1004804), National Natural Science Foundation of China (No. 61702492), Shenzhen Discipline Construction Project for Urban Computing and Data Intelligence, and Shenzhen Basic Research Program (No. JCYJ20170818153016513).

## References

1. Ld, D.B., Krishna, P.V.: Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Appl. Soft Comput.* **13**(5), 2292–2303 (2013)
2. Li, K., Xu, G., Zhao, G., Dong, Y., Wang, D.: Cloud task scheduling based on load balancing ant colony optimization. In: 2011 Sixth Annual China Grid Conference, pp. 3–9. IEEE (2011)
3. Lu, C., Ye, K., Xu, G., Xu, C.Z., Bai, T.: Imbalance in the cloud: an analysis on Alibaba cluster trace. In: 2017 IEEE International Conference on Big Data (Big Data), pp. 2884–2892. IEEE (2017)
4. Ramezani, F., Lu, J., Hussain, F.: Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) ICSSOC 2013. LNCS, vol. 8274, pp. 237–251. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-45005-1\\_17](https://doi.org/10.1007/978-3-642-45005-1_17)
5. Ramezani, F., Lu, J., Hussain, F.K.: Task-based system load balancing in cloud computing using particle swarm optimization. *Int. J. Parallel Program.* **42**(5), 739–754 (2014)
6. Sharma, N.K., Reddy, G.R.M.: Novel energy efficient virtual machine allocation at data center using genetic algorithm. In: 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1–6. IEEE (2015)
7. Sierra, M.R., Coello Coello, C.A.: Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 505–519. Springer, Heidelberg (2005). [https://doi.org/10.1007/978-3-540-31880-4\\_35](https://doi.org/10.1007/978-3-540-31880-4_35)
8. Taylor, S.J., Letham, B.: Forecasting at scale. *Am. Stat.* **72**(1), 37–45 (2018)
9. Yuan, H., Bi, J., Tan, W., Li, B.H.: Temporal task scheduling with constrained service delay for profit maximization in hybrid clouds. *IEEE Trans. Autom. Sci. Eng.* **14**(1), 337–348 (2017)