Cloud Task Scheduling Based on Policy GradientAlgorithm in Heterogeneous Cloud Data Center for Energy Consumption Optimization

CuiHua He YongQuan Yang BoWei Hong
Department of Computer Science and technology
Ocean University of China

三等奖

Abstract

With the widely deploying of large-scale data center, high energy consumption becomes one of the key challenges and difficulties that restricting the development of cloud computing. Due to the limitations and dynamics of resources, how to map the cloud tasks to the resources aiming at decreasing the energy consumption has remained an essential issue. The task scheduling problem is a NP-hard problem , traditionally, various meta-heuristic algorithms can provide a feasible solution under certain conditions.Heuristic algorithm is difficult to adapt to complex and varied cloud environments. The proposed Deep Reinforcement Learning(DRL) has attracted wide attention in recent years and has outstanding ability in solving complicated control problems with high-dimensional state spaces and low-dimensional action spaces, how to apply it to get an efficient task scheduling strategy and make full use of system resource has become a new research focus for researchers.

Proposed Methods

1. Framework

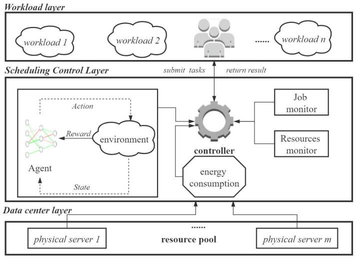


Fig. 1. Cloud task scheduling framework based on DRL

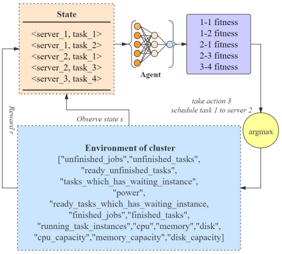


Fig. 2. The design of agent

2. Implements

(1) Introduce a power model to estimate the power of data center of the moment.
(2) Design a reward function that reflects the goal of optimizing power consumption.
(3) Implement the simulation experiment on independent workloads to verity its availability of reducing the energy consumption.
(4) Model inter-task dependencies in jobs and then conduct experiments in a similar way with independent workloads.

3. Energy Consumption Model

$$p_{S_i}^{dy}(t) = p_{S_i}^{st}(t) + (p_{S_i}^{full}(t) - p_{S_i}^{st}(t)) \cdot U_{S_i}(t)^{r_{si}}$$

$$E_i = \int_{t_0}^{t_0+t} p^{type(i)}(u_i)dt$$

4. MDP Model

(1) **State Space**: Fit server-taskInstance list.
(2) **Action Space**: a-th pair in the list is $<s_j,t_i>$ action a represents allocating i-th taskInstance to j-th Server.
(3) **Rewards**: The power of the time step t multiply (-1).

5. Algorithm

(1) Agent Scheduling Algorithm

When make a decision, the agent should choose the pair in the fit server-taskInstance pair list which have the biggest fitness value.If the list is not empty, time will not elapse, agent have to make continuous action until the list is empty and no reward will be given, then an END action will be performed which indicates transfer to the next time step as well as giving the relative calculated reward signal to the agent for learning.

(2) Agent Training Algorithm

In each episode, update the parameters of the agent after N simulations. We use a fully connected neural network to represent the policy πθ(a|s), the purpose of the agent is to maximize toexpectation of cumulative reward Eτ~pθ(τ)[r(τ)].

Algorithm 2: Learning Algorithm of agent

$\alpha=0.001$
N = 15
for each iteration
  for i = 1 to N
    $\tau_i=[s_1^i,a_1^i,r_1^i,\ldots,s_L^i,a_L^i,r_L^i]$
    for t = 1 to L:
      $q_t^i=\sum_{t'=t}^{L_i} r_{t'}^i$
    end for
  end for
  L = max($L_1,\ldots,L_N$)
  for t = 1 **to** L
    $b_t=\frac{1}{N}\sum_{i=1}^{N} q_t^i$
  end for
  for i = 1 **to** N:
    $\Delta\theta=0$
    for t = 1 **to** L:
      $advantage_t^i=q_t^i-b_t$
      $\Delta\theta=\Delta\theta+\nabla_\theta log\pi_\theta(a_t^i|s_t^i)advantage_t^i$
    end for
    $\theta=\theta+\alpha\Delta\theta$
  end for
end for

Experiments

1.Data Set

we use the log data Alibaba Cluster Data V2017 in our experience, the entire user workload is composed of a number of jobs that arrive in an online mode, a job is composed of a number of task and a task has multiple task instances, the task instance is the smallest scheduling unit.

2.Experimental Results and Analysis

(1) Independent Workloads

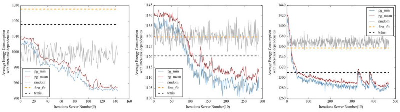


Fig. 3. The average energy consumption of task

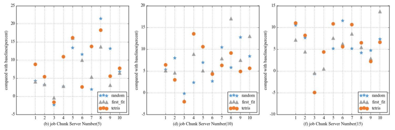
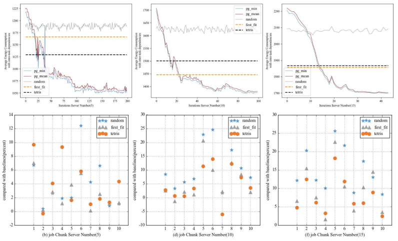


Fig. 4. Percentage reduction in energy consumption

a. At the beginning of the training, the policy gradient algorithm is poor than other algorithms until the training episodes occurs more than a certain number of times, and from that point, the curve gradually converged, and the converged energy consumption is smaller than the three baselines.
b. Except job chunk 3, where the scheduling energy consumption is slightly greater than baselines, other result of job chunk are all less than the baseline, we achieve between 1% to 22 % decrease in energy consumption.

(2) Workloads with Dependencies



The result is similar to independent workloads.

Conclusions

We extends DeepJS for reducing energy consumption in the heterogeneous IaaS cloud environment, The results show that DeepEnergyJS outperforms the baseline algorithms such as RAMDOM, FIRST FIT and TETRIS.

References

[1] Li, Fengcun , and B. Hu . "DeepJS: Job Scheduling Based on Deep Reinforcement Learning in Cloud Data Center." the 2019 4th International Conference 2019.
[2] LiZheng Guo. Research on the Resource Deployment and Task Scheduling in Cloud Computing[D]. Donghua University, 2015: 69-76(in Chinese).

联系方式

联系人：贺翠花
手机：17854287593
邮箱：1577153977@qq.com