2014 IEEE International Conference on Internet of Things (iThings 2014), Green Computing and Communications (GreenCom 2014), and Cyber-Physical-Social Computing (CPSCom 2014)

Approximate Dynamic Programming Based Data Center Resource Dynamic Scheduling for Energy Optimization

Xue Li,Lanshun Nie, Shuo Chen
Research Center of Intelligent Computing for Enterprises&Services
School of Computer Science and Technology, Harbin Institute of Technology
Harbin 150001, China
nls@hit.edu.cn

***Abstract*—As the core part of modern IT infrastructure, data center consumes large amount of energy, which has become the main operational cost. In order to save energy consumption and reduce emission, it's necessary to apply online dynamic scheduling of computational resources and physical resources for division of load, so as to cater for the need of time-variant and random service needs. The paper initiates the layered algorithm for scheduling of data-center resources and establishes energy-consumption models with tractable approximating computations for the data center and, on the basis of approximation dynamic programming method, establishes dynamic scheduling models of large-size heterogeneous resources and the algorithm for learning-based dynamic scheduling of resources. In order to evaluate the fidelity and efficiency of the models, the EnergyPlus and GreenCloud software are integrated into an analogue platform where simulation experiments are conducted and prove the efficiency of the model and the algorithm.**

Keywords-Data center, ADP, CPS, Resource Schedule, Energy Consumption

I. INTRODUCTION

With the boom of IT industry, data center, as an integrator of communication technology and computer and internet, is no longer a colossus, but an arena of competition for IT giants. Rapid development of data center over the years has brought about substantial increase of energy consumption in the data center, which has been prominent as the biggest threat to power supply ability. Therefore, energy-saving is currently a key problem that requires a solution urgently. In view of this, main power-consuming facilities in the data center have become the focus of our research. Because the types of resources in the data center are relatively large in number and in scale and there exists an uncertainty of the users' requirement, if the resources are not properly controlled and scheduled, it will lead to waste of power energy. Hence, the dynamic scheduling of data-center resources, with the goal to optimize energy consumption, is the main research content in this paper. The research, focusing on the current status of high energy consumption of data center, proposes the data center dynamic resource scheduling issue based on Cyber-physical platform, which, while satisfying external needs, adopts an approximation-dynamic programming method for a rational scheduling of computational resource and physical resource, thus reducing energy consumption of data center and maximizing the profits of data center within a given period of time. The paper addresses the important and timely problem of reducing the energy consumption of large data centers by optimally allocating computational and physical resources.

In recent years, researches have been conducted by experts from around the world aiming at the issue of resource scheduling for data center. While for the same problem, this paper makes the following contributions:

- carry out an analysis over the data-center dynamic scheduling issue ;
- suggest a solution based on Cyber-Physical dual layer structure and propose an algorithm for solving the dynamic scheduling problem of data center;
- propose a simulator, it can provide an overall simulation of the energy consumption of the data center.

This paper is organized as follows: Section II summarizes the related work in resource scheduling for data center and uses an example scenario to present challenges in dynamic resource scheduling in data center; Section III models data center's energy consumption and resource; Section IV describes data center resource dynamic scheduling algorithm based on approximation programming; Section V provides the simulation experiments and the analysis of results. Section VI concludes this paper.

II. MOTIVATION AND RELATED WORK

A. Data center resources example scenario

To more clearly describe the DC (Data Center) resource dynamic scheduling problem, an actual case is applied to illustrate its scene, where there is a data center with *n* sets of cabinets, each installed with *m* sets of servers classified into *A, B, C* three types lining *A>B>C* in terms of task processing capacity. Virtualization is realized among servers, which allows load transference and load balancing. The server room is equipped with *p* sets of air-conditioners and *h* sets of ventilators, with the indoor temperature maintaining at *22~24 ℃* and moisture maintaining at *50%~60%RH*. When the outdoor temperature is lower than *10 ℃,*the ventilators can be switched on. In the course of working, the servers mainly processes the Web load whose requests arrive at random and are divided into *high, mid,* and *low* as per priority. Requests of different priorities will be distributed to corresponding servers which are defined as *type-A*, *type-B*, and *type-C* servers in the paper. If the current type of server

DOI 10.1109/iThings.2014.87

is not available, the requests can be directed to other types of servers. When the Web load tasks being processed in the data center are of low volume, some servers, air-conditioners, and ventilators may be switched off according to requirement, for the purpose of satisfying service need and energy saving.

As per the above contents, decision information can be withdrawn as: Within a certain period of time, which server(s) need to be switched on or switched off? How will switch-on and switch-off of the air-conditioner be controlled? How Web load will be distributed to the server? What type of server will it be distributed to?

If DC resource dynamic-scheduling problem is considered as a system, then resource conditions of certain period of time will direct to the status of the system in that period of time, with each period of time corresponds to a scheduling decision, and different decisions will lead to different status. Therefore, values of status may vary with the scheduling strategies, thus reflecting advantages and disadvantages of each scheduling strategy.

B. Challenges

According to the descriptions above, we can summarize the following features and challenges for scheduling of the data-center resources:

External demand and tasks are time-varying, random and periodic; resource characteristics include large-scale, multi-attribute, heterogeneous; the process of solving problem is online.

C. Related Work

In recent years, researches have been conducted by experts from around the world aiming at the issue of resource scheduling for data center. Zahra Abbasi [1] forwards the TACOMA, i.e. the mode of one data center with dual-layer management, which realizes a coordinated dispatching of the dual-layer mode in purpose of reducing the energy consumption of data center to the minimum. For the problem of high power consumption of data center, Michele Mazzucco, Dmytro Dyachuk [2] presents the Platform-as-a-Service mode, which performs well in managing the server cluster and allows the server to be let to users while maintaining the greatest benefits, thus dissolving the confliction between the maximum needs of users and the need for lowest energy consumption. Shuyi Chen, Kaustubh R.Joshi [3] propose a new approach, in any non-virtualized environment or virtualization circumstances, application-aware scaling data center resources to minimize energy consumption. Luca Parolini [4] views data center as a CPS system, his paper establishes computer network mode and hot network mode, and raises a corresponding dispatching strategy by coupling the two modes. E.Le Sueur [5], A.Gandhi [6], L.Rao [7], G.Chen [8] et al. also contribute different solutions in the light of obtaining a balance between maximizing benefits and minimizing energy consumption of the data center.

Based cooling system, reference [9] introduces a data center environmental control system that utilizes a distributed sensor network to manipulate conventional Computer Room Air Conditioner (CRAC) units within an air-cooled environment. Justin Moore [10] proposes and evaluates a simple, flexible method to infer a detailed model of thermal behavior within a data center from a stream of instrumentation data to resolve obstacle to the practical implementation of thermal load management, which is the ability to predict the effects of workload distribution and cooling configurations on temperatures within a data center enclosure. Reference [11] shows through formalization that minimizing the peak inlet temperature allows for the lowest cooling power needs. Using a low-complexity linear heat recirculation model, we define the problem of minimizing the peak inlet temperature within a data center through task assignment (MPIT-TA), consequently leading to minimal cooling-requirement.

III. Problem Formulation

A. Scheduling Method Based on Cyber-Physical Dual-Layer Structure

According to data center’s heterogeneous resources, where IT facilities(servers, storage, routers, etc) of the information layer are called computational resources; while infrastructure such as cooling equipment, lighting equipment, power supply and distribution facilities, etc are called as physical resources. In the light of such, data center can be regarded as a CPS (Cyber-Physical System). As a result, the DC resource dynamic scheduling problem must be resolved with consideration to Cyber and Physical factors for a coordinated dispatching.

What the data center mainly solves is the Web load problem, in which, the web load is viewed as the information entering the system, i.e. the system status is determined by the decision incurred with the arrival of the web load (information). The Web load has a feature of periodical fluctuations, which can be divided into long-period fluctuation and short-period fluctuation. Because frequent on and off switching of air-conditioners and servers may cause greater energy consumption, frequency adjustment must not be applied too frequently, and status of physical-resource equipment needs to be adjusted according to the web load’s long-period fluctuation, for which, we call this layer the Physical Layer. While treatment of the load requires time-effectiveness and short period, and thus needs frequent adjustment, in which the load must be timely distributed due to existence of short-period of fluctuations, resulting in the layer being called as the Cyber Layer. The Cyber-Physical layer structure is demonstrated in Fig. 1.

B. CPS-DE Model (based Cyber-Physical Structure Data-center Energy-consumption model)

Data-center energy consumption is the core concern of this paper, in which an energy consumption model is established based on linearized data center and according to the principle of “close to reality with low complexity”. As mentioned in Section 2, the data center resource scheduling can be divided into the Cyber Layer and the Physical Layer. So respectively, the modeling can also be divided into Cyber Layer, where server energy-consumption is mainly considered, and Physical Layer, where energy-consumption of the cooling system is mainly considered.

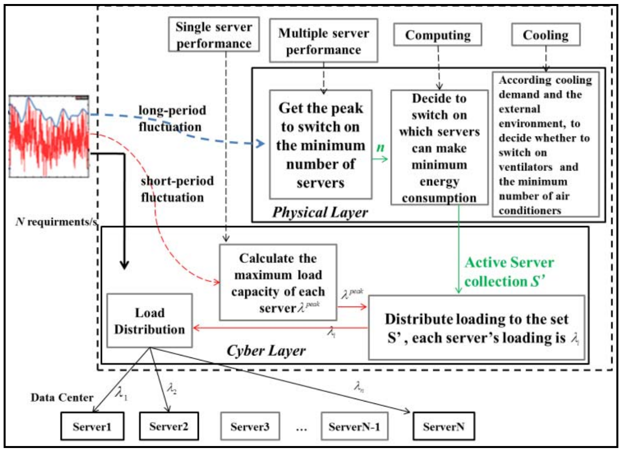

Figure 1. Cyber-Physical Dual-Layer Scheduling Structure

1) Sampling Time Model

The time model in this paper adopts the discrete-time model. As mentioned in the previous sections, the data-center scheduling model requires a dual-layered Cyber-Physical structure. Then the time model is accordingly is as in Fig. 2, in which "T" means a long period, and "t" means a short period.

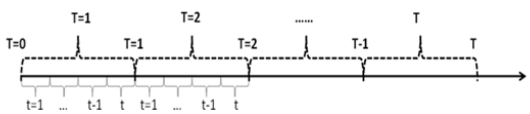

Figure 2. Time model

2) Computation Resource Model

According to investigation, the server energy consumption formula can be expressed as:

$$E_{\text{server}} = \sum_{s=1}^{N}\left(B_s + \alpha U^2 + \beta U\right), \quad (1)$$

N means the number of servers, B_s means the energy consumption of the server S switched on with zero load, and U symbolizes the load rates of the server.

3) Physical Resource Model

a) Energy-consumption model of air-conditioners and ventilators

Air-conditioners work to remove the heat of the data center that comes mainly from two resources: thermal load of indoor equipment (mainly comes from the server's heat load) and the thermal load of environment. On such basis, the general energy-consumption of air-conditioners can be defined as follows:

$$E_{\text{CRAC}} = \lambda\left(E_{\text{server}} \times \eta + \text{K} \times \text{S}\right) + \sigma . \quad (2)$$

Where, S means the area of DC, λ, η, σ, K are coefficients. Similarly, we established ventilators energy consumption model.

Thus, the total energy consumed by air-conditioners and ventilators can be expressed as:

$$E_{\text{Refrigeration}} = E_{\text{CRAC}} + E_{\text{HeatExchanger}} = \left(\lambda - \varsigma o \Delta_{\text{Tem}}\right)\left(E_{\text{server}} \times \eta + \text{K} \times \text{S}\right) + \upsilon \ .(3)$$

Where, Δ_{Tem} means temperature difference between indoor and outdoor.

b) Energy-Consumption Compensation Calculation Model Based on Hot Spot

Data-center servers, due to uneven distribution or fault, may result in uneven exchange of indoor heat and cause hot-spot problems, which can be classified into global server hot spots and cabinet server hot spots.

Global server hot spots: if the working servers are gathered all in one certain area, uneven distribution of heat in the air may easily occur, which may result in what are called global server hot spots. Fig.3 shows the comparison of (a) and (b): the total number of open servers is the same, but (a) is more concentrated the switch-on server, hot forming of the two regions, and (b) is the more evenly distributed. Cabinet server hot spots: when cool air transfers to the remote-control cabinets, or above the cabinet, the wind volume has already weakened down, which may lead to what are called cabinet server hot spots.

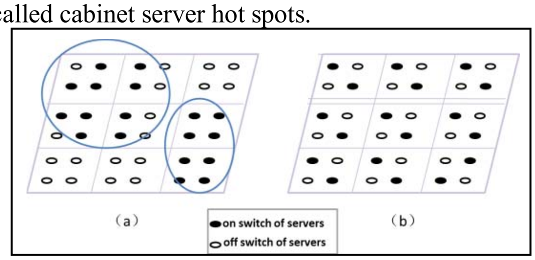

Figure 3. Comparison among Divergence of Global Servers

In view of the above problem, it's necessary to modify the aforementioned formula through simplification regarding the principle of approximation rather than precision, for obtaining an optimized solution, from which, the energy-consumption model based on hot spots can be expressed as:

$$E_{\text{hotSpot}} = \kappa \text{N} + \delta \text{D} + \mu \text{R} + \rho \, . \quad (4)$$

Where, N means the number of servers, D means the divergence of global servers, R means the divergence of cabinet servers; and κ、δ、μ、ρ are coefficients. Due to space limitation, the definition of D, R is not stated.

From the above analysis, the modified data-center energy consumption can be expressed as:

$$\text{Energy} = E_{\text{server}} + E_{\text{Refrigeration}} + E_{\text{hotspot}} \, . \quad (5)$$

C. DC Resource Dynamic Scheduling Model

ADP (Approximate Dynamic Programming)[12] is an effective method for solving nonlinear optimization scheduling of complex systems, which combines stochastic approximation methods and Markov decision processes and other knowledge to solve large-scale resource allocation problem. ADP uses "optimization-simulation" and the value of iterative methods to solve the "dimension of the disaster" .

1) Sampling Time Model

As mentioned in the previous sections, the data- center scheduling model requires a dual-layered Cyber-Physical structure. Then the time model is accordingly T means a long period, and t means a short period.

2) Resource- Utilization Model

Resources in the data center mainly include servers, air-conditioners, and ventilators. How to schedule these three types of resources is the core issue of this paper. Modeling for resources is the priority task in solving the scheduling problem. We establish the Server-use model, air-conditioners and ventilators' model. Because of the limited length of paper, we only describe "server-utilization model".

In modeling for use of servers, important attributes of the servers should be abstracted first in order to make use of the attribute vector for depicting the state variation of servers.

S^R = the set of types of servers={ s_1 =A-type server, s_2 =B-type server, s_3 =C-type server };

s^A ={ server's attribute vector }={ a_1 = server's S.N., a_2 = Server's type, a_3 = server's location, a_4 = server's switching status, a_5 = server's basic energy consumption, a_6 = server's full-power energy consumption, a_7 = server's current load, a_8 = server's processing capacity (processed load volume/second) };

A ={ server's possible attribute vector };

S_{T-a} = the number of servers with the property of a available for scheduling within the long period;

$S_{T-T'-a}$ = the number of servers with the attribute available for scheduling at time T', obtained within the long period T ;

$S_{T,t-a}$ = variation of the number of servers with attribute a caused by external information and obtained within short period t of the long period T , and appearing at time t of the last short period, of which the external information includes: server failure, failure recovery, etc.

3) Load Model

As the task of the data center is to process the load request information which can be regarded as the external information. In the paper, the modeling for the load information is mainly done in the form of Web load, in that the server number in previous resource-utilization model is determined by the maximum load value (peak load value) of the current period. Therefore, when every long period commences, predictions need to be made on the peak load value in accordance with historical data. Load task model needs to depict multiple attributes of load tasks, both static and dynamic, and apply the attribute vectors to depict the status of the tasks while elaborating on the constraint rules for task implementation. The load modeling form is defined as follows:

L^C =set of load task types={ h = load of top-priority, m = load of medium-priority, low = load of bottom-priority};

A_T = {Load attribute vector at the T long period}={ λ^{f-peak} =total load predicted value at peak, λ^{f-h} = predicted value of h-type load at peak, λ^{f-m} =predicted value of m-type load at peak, λ^{f-l} =predicted value of l -type load at peak, $\overline{\lambda}^f$ =average load predicted value, $\overline{\lambda}^r$ =average actual load value, $\underline{\lambda}$ =actual processed load};

$A_{T,t}$ ={load attribute vector at the t short period of the T long period value}={ λ =total load, $\lambda^h = h$ -type load, $\lambda^m = m$ -type load, $\lambda^l = l$ -type load, s =load processing status };

$M_{T,t-T',t'-a}$ = load of attribute a obtained at the t short period of the T long period and to be implemented before the t' short period of the T' ;

$M_{T,t-a}$ = load of attribute a obtained at t short period of T long period and to be implemented at current time.

4) Decision Model

In the modeling theory of the paper, decision belongs to the internal information of the system. Decision model is established in order to depict how the decision acts upon the load tasks and the data center resource, and how it causes change of the system status. Therefore, modeling for scheduling decision needs to begin with consideration of its influence on data center resource and load tasks, abstract important attributes from them, and define the decision sets and the constraints on vector space value-setting. This paper elaborates the idea of model only, if you want to know the detail information, please contact us.

In order to reflect the decision result in form of mathematics, it's necessary to define a decision function and some scheduling strategies, and provide the system's status information at each sampling moment, and returns for scheduling decision.

$X_T^\pi(R_T)$ = decision function, expressing that a decision vector x_t is returned at T long period, under scheduling decision π and the R_t status of the data-center resources, in which R_t is the status information of the data-center resources at time T.

Where,

$$X_t^\pi(R_{t-1}) = \arg\max_{x_t \in \chi_t}\{C_t(R_{t-1}, \omega_t, x_t) + \gamma \bar{V}_t\ (R^M(R_{t-1}, \omega_t, x_t))\}. \quad (6)$$

scheduling decision is jointly reflected by contribution function (reflecting what contribution current scheduling decision has for the system), approximation value function (reflecting the influence current scheduling decision has on the future of the system). x_t means the decision attribute vector; $\hat{v}_t^n$ represents the digital expectation of approximated values transferring from S_t to S_{t+1} ; $R^M(R_{t-1}^n, \omega_t^n, x_t) = R_t^n$ indicates the status transferring function of the data center; $C_t(R_{t-1}^n, \omega_t^n, x_t)$ refers to the system's contribution function; and $\bar{V}_t^{n-1}(R_t^n)$ means the approximated value function.

On dynamic scheduling of data-center resources, different costs may be needed for scheduling of unused resources, and different revenues may be obtained for processing different tasks of load requests. According to different servers' energy consumption and different revenues from processing of different load tasks, relevant revenue weights may be obtained. Cost in this paper represents the energy consumption of the data-center resources.

For each sampling time, the contribution function can be defined as:

$$C_t(S_t, x_t, W_{t+1}) = total(revenue) - total(\cos t) . \quad (7)$$

If you want to know the detailed solution process, please email to us.

5) Objective Equation

It is mentioned that the data-center scheduling problem is considered as a dynamic system, in which each scene of scheduling decision means a status S_t of the system. Through the resource status R_{t-1} of the data center, S_t is described jointly by load-requesting information M_t and decision x_t . S_t's value is jointly demonstrated by contribution function and approximation value function. The optimized objective of the problem is: to bring the energy consumption to the lowest while satisfying the needed load within a certain period of time. The long-term objective is: to maximize the profits of the data center while allowing it to bear as much load as possible. This can be expressed by system's status value as the optimal scheduling decision selected at every setting time, and will be able to maximize the system's status value. Then, the objective equation for the DC resource dynamic scheduling problem can be expressed as:

$$\hat{v}_t^n = \max_{x_t \in \chi_t} C_t(R_{t-1}^n, \omega_t^n, x_t) + \gamma \bar{V}_t^{n-1}(R^M(R_{t-1}^n, \omega_t^n, x_t)) \cdot \quad (8)$$

for which the solution is the decision attribute vector x_t .

IV. ALGORITHM FOR DYNAMIC-SCHEDULING OF DC RESOURCES BASED ON ADP

1) Algorithm Design

The solving process based on data-center resources dynamic scheduling problem can be divided into two periods: the first period is to train the historical data for obtaining the approximation value function, and the second period is to apply the expression formula of approximation value obtained from the training in the first period for directing the scheduling of data center resources. Scheduling of data center resources is done on Physical layer and Cyber layer, with data-center resources scheduling problem focusing mainly on Physical layer, and the computational problem of load distribution focusing mainly on Cyber layer. Therefore, the ADP method is employed only to the Physical layer in the paper.

In the training period, first assume the total number of sampling is *N*, and the sampling route(based on the time continuity, according to external information, deciding a continuous state with the same cycle length) length is τ , give the approximated estimated initial value $\bar{V}_t^0$ of the corresponding system status at each sampling moment and at each sampling route, and then apply iterative computations to (8) via method of value iteration. In each process of iteration, the system status value $\hat{v}_t^n$ obtained via computation at the sampling route ω^n is used for updating the approximated estimated value $\bar{V}_{t-1}^n$ of the previous system status, and thus constantly approximating the real value of the system status. Finally, N groups of approximated values $(\bar{V}_t^n) = \{(\bar{V}_t^1)_{t=1}^{\tau}, (\bar{V}_t^2)_{t=1}^{\tau}, ..., (\bar{V}_t^N)_{t=1}^{\tau}\}$ can be obtained for the system status, from which, the group of values reaching steady state can be chosen for linear fitting and for acquiring the expression formula of approximated value function.

Then in the training period, the input of algorithm is the historical load information, and the output of algorithm is approximated value of the system status in each of the training period. In the paper, simulation historical task is used for obtaining and measuring some parameters of the algorithm at the training period, such as discount factors, system status initial value, step size, etc.

The application period is to apply the approximated value function obtained in the training period, and seek solution in the light of current information of data-center resource for obtaining the optimized scheduling decision x_t . Therefore, in the application period, the input of algorithm is the task information of load request, and output is the optimized scheduling decision x_t .

2) Heuristic Rule of Scheduling Decision

a) Heuristic Rules of Resource Scheduling Decision Based on Long Period

Algorithm 1 Heuristic rules for scheduling data-center resources at long periods

Step1: Generate randomly the load peak value at the current long period *T* according to the historical data, and calculate the high, mid, and low types of load for calculating the minimum number of each type of servers needed to be switched on.

Step2: Observe in each type of servers if there is any one not in full load, and if the load can be combined so that some server(s) may be switched off to reduce the number of servers to the minimum. If the load can be combined for switching off certain set of server, it's recommended to switch off the server with the lowest capacity of computation. Then, the total number of servers *N*, and the number of each type of servers.

Step3: Check the current status of the servers, and determine the number of servers needed for adjustment, A^S, B^S, C^S .

Step4: if $A^S + B^S + C^S < \sigma$, where σ is the adjustment rate, $\sigma = N_{T-1} / N_T$

do there is no need to adjust the number of servers

else

do after adjustment of A^S, B^S, C^S is finished, a set *S'* of active servers is obtained

Step5: Judge on the external temperature T_{out} . If the $T_{out} < \theta$,then the ventilators switches on, θ is the critical temperature when at switching on of the ventilators.

Step6: Determine the minimum number of air-conditioners needed to be switched on for cooling according to the number of active servers, if the ventilators are switched on, and the difference between indoor temperature and outdoor temperature.

Step7: If there is any server switching off, then transfer the load unimplemented on this server to other servers.

Step8: In the process of adjustment, calculate, according to the definition of contribution function, the contribution values corresponding to scheduling decisions of different server's position, and carry out position scheduling for the servers according to current maximum contribution value corresponding to the scheduling decisions.

Step9: **for** *t=1,2….t* (each short period)

Assign the load to active servers according to the heuristic rules of load distribution.

Step10: Current load, if unimplemented during this period, will be transferred during the next period according to **Step1**

b) Heuristic Rules Based on Short-Period Resource-Scheduling Decisions

Algorithm 2 Heuristic rules for short-period load distribution

Step1: Put the load unimplemented in the previous short period together with the load arriving in current short period into the implement list.

Step2: Check if there is load waiting for over the setted time, for example *10s*; if so, implement such load first.

Step3: Implementation of other loads shall be performed as per load types and the time order of arriving of the loads

Step4: Distribution method: generate random numbers, and assign the load to be distributed to the servers corresponding to the random numbers. If the current type of servers is full-loaded, then the load will be distributed to the servers satisfying the conditions with the lowest difference in performance from current type of servers.

Step5: Implement the load

Step6: The load unimplemented will be extended to the next short period for implementation.

3) Training and Applying Algorithm

In the training period, the algorithm will mainly adopt methods of value iteration and smooth strategy for achieving the real value of system status.

Algorithm for application period is used to apply approximated value function obtained in the training period. Assume the linear expression formula for the approximated value function as:

$$\bar{V}_t^n = \theta_1 FL + \theta_2 S_A + \theta_3 S_B + \theta_4 S_C + \theta_5 L + \theta_6 C + \theta_7 \cdot \quad (9)$$

where FL is a predictable load peak value; S_n is the number of active *n-type* servers; L is the dispersity of servers' position; and C is the number of air-conditioners in running.

Because of limited space, we couldn't describe these two algorithms in detail.

V. Implementation And Evaluation

1) Training Experiment

In the light of the above algorithm, relevant experiments are conducted relating to the problems defined in Section II. The experiment assumes there are three types of servers in a server room of *600* ㎡, with each type of servers equipped with *383* sets of servers, *12* sets of air-conditioners, and *4* sets of ventilators. The value iteration method requires a rational convergent criterion.

Due to the periodicity of Web load, in order to make the simulation data more close to real, a period is assumed as a sampling route; set each two hours as a long period unit, for which every two minutes is set as a short period unit.

According to the periodicity of Web load, in the experiment, each sampling route generated randomly will correspond to the predicted load peak value and the *high* load, *mid* load and *low* load respectively under the peak value. The task load generated during each period is uncertain, and the random and uncertainty problem of load tasks used for simulating the scheduling course results in the dynamicity problem.

Rational convergence is needed in adoption of the ADP algorithm for training in the course of long-period scheduling. In the experiment of the paper, sampling is performed respectively for *50* times, *120* times, *150* times, and *180* times for observation whether system's status value on each sampling route is steady as compared with the previous moment, in which *50* times of sampling shows a complete misconvergence, *120* times of sampling shows a trend of convergence, *150* times of sampling shows a good convergence, and *180* times of sampling shows a steady state. This indicates that application of value iteration strategy has already been able to approximate the true value and reached a good state of convergence, and can terminate. As shown in Fig.4.

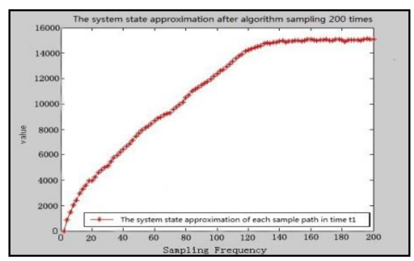

Figure 4. The Value Tends to Converge after 180 Times of Iteration by the Algorithm

Select the *84* sets of values after the convergence becomes stable, and seek linear solution for the approximation value function. Linear regression calculation produces the solution of the coefficients θ_1 etc in (9), and the linear expression formula of this approximation value function can be expressed as (10):

$$\bar{V}_t^n = -0.24372FL + 38.03436S_A + 26.25947S_B + 1.113749S_C + 3841.669L + 20.98863C + 1026.15 \cdot \quad (10)$$

2) Application Experiment

a) Simulator

For the data center, there hasn't been software aiming at simulating total energy consumption of the data center. In consideration of the synergy degree between computational resources and physical resources, an overall energy consumption simulation software is needed that can simulate the energy consumption for computation and reflect the physical energy consumption while satisfying the requirements. Therefore, the paper proposes to integrate the EnergyPlus [13] software that simulates energy consumption of computation resources and the Green Cloud Simulator [14] that simulates the energy consumption of physical resources, so that the integrated software may provide an overall simulation[15] of the energy consumption of the data center. This integrated software is called integral data-center energy-consumption simulation software. Its interaction principle shown in Fig. 5 .

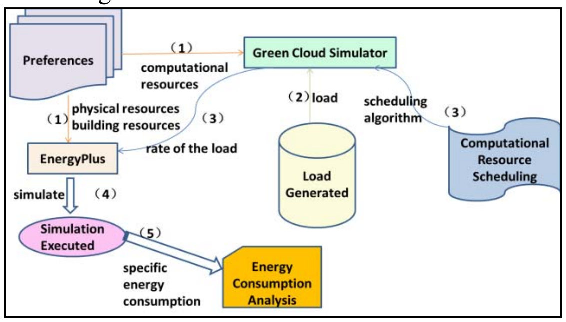

Figure 5. EnergyPlus and GreenCloud 's Interaction Principle

b) The Fidelity of CPS-DE Model

The "CPS-DE model" is close to real practice and easily solved. Because the parameters obtained for "CPS-DE model" are fittings calculated via EnergyPlus, similar to the fitting method applied in approximation value function in Section5. Therefore, it's necessary to compare "CPS-DE model" with actual situation.

In the process of obtaining solutions of parameters for the energy-consumption function, *200* sets of experimental data is drawn out and calculated via EnergyPlus software for obtaining the simulated energy consumption, and then seeking solutions via the linear regression method in the Excel. The next step is to carry out verification on the fidelity of the "CPS-DE model". Take another *200* sets of experimental data, use the "CPS-DE model" function and the EnergyPlus software respectively for obtaining the energy-consumption values, the results of which are shown in Fig. 6. From the tendency of the function, it can be seen that the "CPS-DE model" tendency and actual tendency are generally proportionate. Though there do exist some derivations, they don't affect the directing on optimization of the function. Via comparing these *200* sets of data, of the calculations done by the function model, the ones matching with actual cases reach 92.5%, and the ones not reach 7.5%, which proves the "CPS-DE model" is reliable to certain degree.

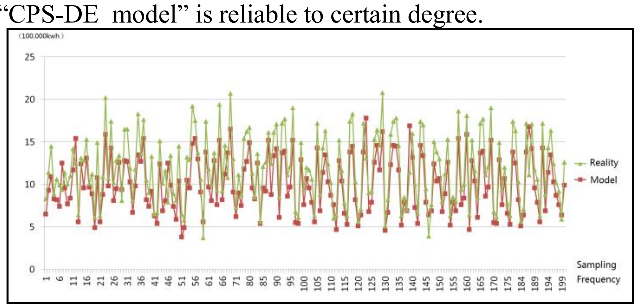

Figure 6. The "CPS-DE model" and Actual Situation's Comparison

b) Application of Simulation Experiment

Firstly, set parameters of the data center which must be in line with those mentioned in the scene of training experiment. Obtained approximation value function is correlated to the scene of the training experiment, and different scenes may result in different approximation functions. Therefore, when applying the approximation value functions, the scenes must be kept consistent. Then, predict the Web load. Due to periodicity and similarity of the Web load, one period can serve as a sampling period, i.e. on a sampling route by ADP method for description of the experiment. Then, according to the predicted load, schedule the approximation value function for solution of an optimal scheduling decision, based on which the energy consumption of servers in every region can be calculated, and so can the switching-on power of energy facilities in the region. In this way, the time table can be designed for differentiating service days and non-service days. This is the case of a single sampling route, to which other sample routes are similar. Finally, conduct an analogue simulation on the data-center models established by EnergyPlus and the energy consumption in this case can be obtained.

3) Performance of the Algorithm

In order to evaluate the advantages of the algorithm, an analysis should be performed first on the strategy of the algorithm. The advantages of ADP-based algorithm for dynamic scheduling of data-center resources are mainly demonstrated in that the algorithm considers not only the influence of current decision on the current system status in the scheduling decision at every long period, it also considers the influence of current decision on the system's status in every future periods. Therefore, in this paper, only the influence of current decisions on the system in the current period, rather than in future periods, is considered, and a greedy algorithm is designed for dynamic scheduling of data-center resources, which switches on the server according to the predicted load of each current time, and considers no influence on the future periods. Through comparison of the two, it can be seen which kind of scheduling strategy is more advantageous. Via comparing both algorithms, the load is obtained by prediction, and the loads of both algorithms are given as identical.

Table I and Table II respectively show the scheduling course of each of the algorithms. If the electric charge is considered as *¥1* per kilowatt hour, the total profits of the data center under scheduling by each of the algorithms can be obtained through calculation. Table III shows the result of average on *20* samplings over a year's profits of the data center implemented respectively by the two algorithms.

In conclusion, the comparison of greedy scheduling algorithm with ADP-based scheduling algorithm proves in full that ADP-based algorithm for dynamic scheduling of data-center resources is more advantageous the other. Moreover, we compared ADP-based algorithm with predictive ant colony algorithm. The result shows ADP is superior to others.

TABLE I. ADP-BASED DECISION PROCESS

Predicted load	Active server A	Active server B	Active server C	Server divergence	air-conditioners in running
27325	136	175	164	0.286	5
41340	255	175	217	0.325	6
44639	255	237	196	0.347	7
37006	194	210	196	0.159	6
30235	173	165	152	0.122	4
…	…	…	…	…	…

TABLE II. GREEDY ALGORITHM DECISION PROCESS

Predicted load	Active server A	Active server B	Active server C	Server divergence	air-conditioners in running
27325	136	175	171	0.345	5
41340	255	178	241	0.307	7
44639	249	246	236	0.386	8
36007	194	195	203	0.205	6
30235	189	177	144	0.317	4
…	…	…	…	…	…

TABLE III. TWO ALGORITHMS' COMPARISON

Method	ADP	Greedy
Implemented loads(*100 million pieces*)	2703725	2703327
Average revenues (*¥/100 million pieces*)	2.70	2.69
Total revenues(*million¥*)	7.30	7.27
Total energy consumption(*million¥*)	2.95	3.35
Total profits(*million¥*)	4.35	3.92

VI. CONCLUSIONS

In response to the high energy consumption of the data center, the paper puts forward the two-layered Cyber-Physical structure for dynamic scheduling of resources for the purpose of energy-saving and emission-cutting, establishes system's resource model and scheduling decision model on the basis of approximation dynamic programming, and proposes a series of heuristic rules for dynamic scheduling of resources. The experiment shows that the approximation value function can form an effective approximation to the status value, and the scheduling generated by the algorithm (with absence of future information) can cover both the current and future status of the system, obviously better than the greedy algorithm. Therefore, the models and methods are an effective way to solve the problem of dynamic scheduling of data-center resources. In the actual course of application, it's influenced by model quality, parameter recognition, availability and accuracy of training data, and real-time track of the system's evolution states etc, so the real system needs to be adjusted, simulated and approximated for adaptability.

REFERENCES

[1] Abbasi Z, Varsamopoulos G, Gupta S K S, "TACOMA: Server and workload management in internet data centers considering cooling-computing power trade-off and energy proportionality," ACM New York, NY, USA , Vol. 9,pp 11. June 2012.

[2] Mazzucco M, Dyachuk D. "Optimizing Cloud providers revenues via energy efficient server allocation". Sustainable Computing: Informatics and Systems, Vol. 2, pp 1–12.March 2012.

[3] Chen S, Joshi K R, Hiltunen M A, et al. Using CPU gradients for performance-aware energy conservation in multitier systems[J]. Sustainable Computing: Informatics and Systems, 2011, 1(2): 113-133.

[4] Parolini L, Sinopoli B, Krogh B H, et al. "A Cyber–Physical Systems Approach to Data Center Modeling and Control for Energy Efficiency". IEEE, Vol. 100, pp 254-268. Jan 2012.

[5] Le Sueur E, Heiser G. "Dynamic voltage and frequency scaling: The laws of diminishing returns",Proceedings of the 2010 international conference on Power aware computing and systems. USENIX Association,pp 1-8. 2010

[6] Gandhi A, Gupta V, Harchol-Balter M, et al. "Optimality analysis of energy-performance trade-off for server farm management". Performance Evaluation, pp 1155-1171. 2010.

[7] Rao L, Liu X, Ilic M, et al. "Mec-idc: joint load balancing and power control for distributed internet data centers". Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems. ACM, pp 188-197.

[8] Gandhi A, Chen Y, Gmach D, et al. "Minimizing data center sla violations and power consumption via hybrid resource provisioning".Green Computing Conference and Workshops (IGCC), 2011 International. IEEE, pp1-8.2011.

[9] Bash C E, Patel C D, Sharma R K. Dynamic thermal management of air cooled data centers[C]//Thermal and Thermomechanical Phenome na in Electronics Systems, 2006. ITHERM'06. The Tenth Intersociety Conference on. IEEE, 2006: 8 pp.-452.

[10] Moore J, Chase J S, Ranganathan P. Weatherman: Automated, online and predictive thermal mapping and management for data centers[C]// Autonomic Computing, 2006. ICAC'06. IEEE International Conferen ce on. IEEE, 2006: 155-164.

[11] Qinghui Tang , Sandeep Kumar S. Gupta , Georgios Varsamopoulos ."Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Data Centers:A Cyber-Physical Approach" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, Vol. 19,pp 1458.Nov 2008.

[12] Warren B.Powell. "Approximate Dynamic Programming for Operations Research: Solving the curses of dimensionality". Hoboken, New Jersey: WILEY-INTERSCIENCE,2007.

[13] Crawley D B, Lawrie L K, Pedersen C O, et al. "Energy plus: energy simulation program". ASHRAE journal, pp 49-56, 2000.

[14] Kliazovich D, Bouvry P, Khan S U. "GreenCloud: a packet-level simulator of energy-aware cloud computing data centers". The Journal of Supercomputing, pp1263-1283, 2012.

[15] B. Li, Z. Sun, K. Mechitov, G. Hackmann, C. Lu, S. Dyke, G. Agha a nd B. Spencer, "Realistic Case Studies of Wireless Structural Control, " ACM/IEEE International Conference on CyberPhysical Systems (IC CPS'13), April 2013.