

Power cost minimization in data centers Via Lyapunov optimization^①

Zhang Ran (张 然)^{②*}, Wan Jianxiong^{**}

(* School of Mathematics and Statistics of Central South University, Changsha 410083, P. R. China)

(** School of Information Engineering, Inner Mongolia University of Technology, Hohhot 010051, P. R. China)

Abstract

With the advent of big data, the demand for computing has been increasing in a very large scale for the past decade, so geographically distributed data centers are erected in the direction of cloud computing development. A Lyapunov optimization approach is considered for the problem of minimizing energy cost for distributed Internet data centers (IDCs). By capturing the power cost of servers and cooling systems, the Lyapunov optimization technique is formulated to design a decisive strategy that offers provable power cost minimization and QoS guarantees. The algorithm performance and effectiveness are validated via simulations driven by real world traces.

Key words: power cost, quality-of-service (QoS), workload

0 Introduction

The increasing Internet services and cloud computing have stepped into people lives in recent years so that big data and computation are migrated to or hosted on the Internet data centers (IDCs). The total construction area of a huge IDC is 300,000 square meters, which can host more than one million high-performance servers. However, consumed energy in IDCs and its cost have been gradually out of control. Qureshi et al. discovered that many IDC operators consumed more than 10 million dollars on their annual electricity bills^[1], so the research focuses on how to reduce energy consumption and electricity cost of IDCs.

Most previous jobs on power management paid close attention to how to reduce the total energy consumption. However, apart from the energy consumption, electricity price should be more concerned too since the electricity prices in western countries exhibit time and location diversities^[2-5]. Although the above researches probed the energy cost spent in electricity from servers and practical applications, they ignored another aspect of energy cost, cooling system. Zhang designed and evaluated TEstore exploiting thermal and energy storage techniques to cut the electricity bill for data center cooling without causing servers in a data center to overheat^[6]. Nevertheless, the study simply reduced the total cost in a coarse-grain pattern. Their research ignored the widen difference of server-temper-

atures across diverse server racks and applications.

An expense minimization of IDC's electricity power (EMIEP) problem is formulated in this paper to minimize time-averaged expected energy cost subject to QoS and average temperature constraints. Meanwhile, the research designs an algorithm leveraging the Lyapunov Optimization technique to approximately solve the EMIEP problem and use real workload trace from Ordos UniCloud Technology Co., Ltd. to simulate the above algorithm. Numerical results illustrate that the presented algorithm can reduce total energy cost as well as guarantee QoS and temperature constraints.

1 System model

This section discovers a system model shown in Fig.1 and formulates the EMIEP problem. IDC physically consists of racks of servers and is logically made up of a number of applications, each of which schedules its servers to process the arriving service. All requests preparing to enter an application share the same workload queue and follow the rule that the requests in buffer will be scheduled to the currently idle server. Quality-of-service (QoS) requirement must be considered in each application.

The running servers generate massive waste heat inevitably. For the sake of the reliability of the servers, a computer room air conditioner (CRAC) is used to regulate the server temperatures by sucking cold air into the server racks, pushing waste heat out of the ma-

① Supported by the National Natural Science Foundation of China (No. 61502255), the Inner Mongolia Provincial Natural Science Foundation (No. 2014BS0607), and the Science Research Project for Inner Mongolia College (No. NJZY14064).

② To whom correspondence should be addressed. E-mail: seran_zhang@126.com

Received on Oct. 9, 2015

chine room, and recycling the inside air via the air chilling unit. Fig. 1 exhibits the air flow of the system. The light arrow represents cold air and the brunet one symbolizes hot air exhausted from the racks. Apart from the above CRAC, an indoor air conditioner maintains the machine room temperature to predetermined T^{sp} .

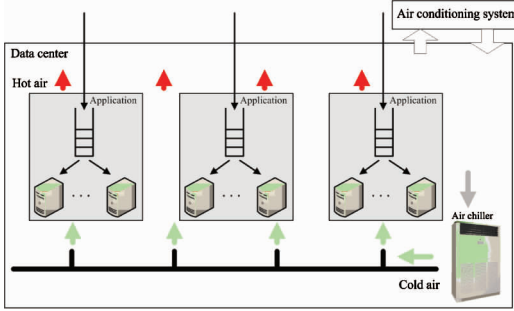


Fig. 1 Layout of IDC Applications

The whole industry is confronted with huge cost of energy consumption, including server energy consumption and cooling energy consumption. This paper formulates the problem of minimizing the total energy cost of the data center subject to QoS and server temperature constraint. The formulated model can be simply described as follows:

minserver power cost + cooling power cost
subject to:

QoS constraint, for each application

Average temperature constraint, for each server

This problem describes three control variables: number of servers for each application, cold air temperature and electricity price discussed in details in the following sections.

2 Problem formulation

2.1 Energy cost model

2.1.1 The server side

Let J be the total number of applications hosted in IDC. At time slot t , define $p(t)$, $e_j(t)$, $L_j(t)$, and $m_j(t)$ as the electricity price, energy consumption for a single server, the workload, and the number of servers for application $j \in [1, \dots, J]$, respectively. Refs[7,8] presented a linear function to display the relationship between power consumption and the server load as follows:

$$e_j(t) = a_1 \frac{L_j(t)}{m_j(t)} + a_2 \quad (1)$$

where a_1 is the marginal energy consumption for CPU and a_2 denotes the server energy consumption except CPU. The total energy consumption of an application j

is $E_j(t) = m_j(t) \times e_j(t) = a_1 L_j(t) + a_2 m_j(t)$, and the total server energy cost in IDC is

$$\begin{aligned} PS(t) &= \sum_{j=1}^J p(t) E_j(t) \\ &= \sum_{j=1}^J p(t) (a_1 L_j(t) + a_2 m_j(t)) \end{aligned} \quad (2)$$

2.1.2 The air chiller side

The energy consumed by the air chiller and drop in air temperature from T_1 to T_2 ($T_1 > T_2$) can be denoted as $\frac{cf\rho(T_1 - T_2)}{COP}$ ^[8], where c is the heat capacity of the air (expressed in Joules/(kg. K) unit), f the air flow rate (expressed in m^3/s unit), the air density (expressed in kg/m^3 unit), and COP the coefficient of performance representing the efficiency of the cooling unit. Without loss of generality, it is usual to assume $COP = 1$. In the machine room, the air chiller should lower the air temperature from the room temperature (T^{sp}) to instant cooling air temperature at time slot t ($T^c(t)$). Hence, the energy consumption is

$$C(t) = cf\rho(T^{sp} - T^c(t)) \quad (3)$$

The energy cost is

$$PC(t) = p(t) \times cf\rho(T^{sp} - T^c(t)) \quad (4)$$

Obviously, it is the sum of $PS(t)$ and $PC(t)$ that equals the total power cost of the data center.

2.2 Constraints

2.2.1 QoS constraint

In consideration of requests in an application of sharing the same queue, a $M/M/N$ queuing model is applied to the near response time. The average queuing delay can be expressed as $\frac{1}{m_j(t)\mu - L_j(t)} \times P_Q$,

where P_Q denotes the probability that the waiting queue is not empty. Without doubt, servers in a running data center are nearly always busy, so $P_Q \approx 1$. For application j , the mean service rate μ_j (expressed by request/second) can be obtained via dividing the speed of the CPUs (expressed by commands/second) by the mean number of commands for a request K_j , i. e., $\mu = \frac{s}{K_j}$.

Given a predetermined average delay upper bound D_j for application j , the QoS constraint can be calculated as

$$\frac{1}{m_j(t)\mu - L_j(t)} \leq D_j, \quad \forall j \quad (5)$$

2.2.2 Temperature constraint

At steady state, the temperature of server j can be controlled by the inlet cold air temperature $T^c(t)$ and the CPU energy consumption $e_j(t)$:

$$T_j^{CPU}(t) = T^c(t) + \sigma e_j(t) \quad (6)$$

where ζ is the heat exchange rate expressed in $K \cdot s/J$. The processor and motherboard reliability of a server is mainly influenced by temperature gradient and thermal stress in a production data center. With respect to reliability issue, the expected server temperature must be maintained below a certain threshold T^{\max} . Plugging Eq. (1) into Eq. (6) yields:

$$E\left\{T^c(t) + \zeta\left(a_1 \frac{L_j(t)}{m_j(t)} + a_2\right)\right\} \leq T^{\max}, \forall j \quad (7)$$

2.3 Problem formulation

After some alternations of Eq. (5) and inequation(7), now define the EMIEP problem as follows:

$$\min \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T E\{PS(t) + PC(t)\} \quad (8)$$

subject to:

$$\frac{\frac{1}{D_j} + L_j(t)}{\mu} - m_j(t) \leq 0, \forall j, t \quad (9)$$

$$E\left\{L_j(t) - \left(\frac{T^{\max} - T^c(t)}{\zeta a_1} - \frac{a_2}{a_1}\right)m_j(t)\right\} \leq 0, \forall j, t \quad (10)$$

Finally some bound constraints for the decision variables are added:

$$m_j \leq m_j^{\max} \\ T^{\min} \leq T^c \leq T^{\max}$$

The EMIEP problem cannot be easily solved due to the following reasons: 1) The unknown probability distribution of $L_j(t)$ and $p(t)$ makes the expectation in Eq. (8) and constraint (10) computationally troublesome; 2) Traditional methods to deal with dynamic optimization problems such as dynamic programming suffers from the curse of dimensionality, i. e., the computation complexity grows exponentially with problem size. Therefore, a modern method will alternate the old approach to approximately solve this problem.

3 A Lyapunov approach to solve EMIEP problem

This section first relaxes constraint (10) and uses the Lyapunov optimization theory to obtain an optimal solution.

3.1 Relaxing the EMIEP problem

Constraint (10) can be relaxed into

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T E\left\{L_j(t) - \left(\frac{T^{\max} - T^c(t)}{\zeta a_1} - \frac{a_2}{a_1}\right)m_j(t)\right\} \leq 0, \forall j \quad (11)$$

The relaxation suggests that the expected server temperature is occasionally beyond the temperature bound without destroying reliability as long as the time-

averaged expected temperature is within the acceptable range.

Replacing constraint (10) in the original problem by constraint (11) leads to an relaxed version of EMIEP. Constraint (11) can be further transformed into:

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T E\{m_j(t)T^c(t) + \zeta(a_1 L_j(t) + a_2 m_j(t)) - T^{\max} m_j(t)\} \leq 0 \quad (12)$$

To satisfy constraint (12), EMIEP can be transformed into a queue stability problem. Define virtual queue $Z_j(t)$ with update Eq. (13).

$$Z_j(t+1) = \max\{Z_j(t) + m_j(t)T^c(t) + \zeta(a_1 L_j(t) + a_2 m_j(t)) - T^{\max} m_j(t), 0\} \quad (13)$$

Then, if $Z_j(t)$ is mean rate stable, i. e.,

$$\limsup_{T \rightarrow \infty} \frac{E\{Z_j(t)\}}{T} = 0, \text{ constraint (12) is satisfied.}$$

3.2 The objective function

The Lyapunov optimization framework subtly designs a control algorithm that chooses actions for all t to yield a time average expectation of the objective function value close to optimal solution with the mean-stable virtual queue $Z_j(t)$. The algorithm changes the original problem into an alternative, minimizing the time average of a cost function subject to queue stability.

Let $Z_j(t)$ be a concatenated vector of all virtual queues with update Eq. (13). Define the Lyapunov function:

$$L(Z(t)) \triangleq \frac{1}{2} \sum_{j=1}^J Z_j(t)^2 \quad (14)$$

Define $\Delta(Z(t))$ as the conditional Lyapunov drift for slot t :

$$\Delta(Z(t)) \triangleq E\{L(Z(t+1)) - L(Z(t)) | Z(t)\} \quad (15)$$

where the expectation depends on the control policy and random workload arrivals.

Instead of taking control actions to directly minimize Eq. (8), the Lyapunov optimization seeks to minimize a bound of the following drift-plus-penalty function

$$\Delta(Z(t)) + VE\{p(t)(E(t) + C(t)) | Z(t)\} \quad (16)$$

where $V \geq 0$ symbolizes an "importance weight" on how much the algorithm emphasizes cost minimization. In section 6, an algorithm minimizing Eq. (16) will achieve a close-to-optimal solution while stabilizing $Z(t)$.

The drift-plus-penalty objective function Eq. (16)

can be bounded by the following

$$\begin{aligned} \Delta(\mathbf{Z}(t)) + \text{VE}\{p(t)(E(t) + C(t)) \mid \mathbf{Z}(t)\} \\ \leq B + \sum_{j=1}^J Z_j(t) E\{(m_j(t)T^c(t) \\ + \zeta(a_1 L_j(t) + a_2 m_j(t)) - T^{\max} m_j(t)) \mid \mathbf{Z}(t)\} \\ + \text{VE}\{p(t)(E(t) + C(t)) \mid \mathbf{Z}(t)\} \quad (17) \end{aligned}$$

where B is a constant defined as $B = m^{\max} T^{\max} + \zeta(a_1 L^{\max} + a_2 m^{\max})$. The designed algorithm minimizes the right-hand-side of Eq. (17).

3.3 Algorithm design

The objective function of problem Eq. (8) can be rewritten as

$$\begin{aligned} \min \sum_{j=1}^J Z_j(t) [m_j(t)T^c + \zeta(a_1 L_j(t)) + a_2 m_j(t) \\ - T^{\max} m_j(t)] + V \sum_{j=1}^J p(t) [a_1 L_j(t) + a_2 m_j(t) \\ + cfp(T^{SP} - T^c)] \quad (18) \end{aligned}$$

The existing cross term of control variables plagues the solution of the problem. However, fixing $L_j(t)$ and $p(t)$ can simplify the solution for $m_j(t)$ by creating a linear function. To see that, dropping the constant terms in Eq. (18) yields

$$\begin{aligned} \sum_{j=1}^J \{Z_j(t)(m_j(t)T^c + \zeta a_2 m_j(t) - T^{\max} m_j(t)) \\ + V_p(t) a_2 m_j(t)\} - V_p(t) cfp T^c \quad (19) \end{aligned}$$

Rearranging (19) yields

$$\begin{aligned} \sum_{j=1}^J m_j(t) \{Z_j(t)(T^c + \zeta a_2 - T^{\max}) + V_p(t) a_2\} \\ - V_p(t) cfp T^c \quad (20) \end{aligned}$$

If T^c is fixed, $m_j(t)$ can be solved analytically: if the coefficient of $m_j(t)$ is greater than 0, then other-

wise $m_j(t) = \frac{\frac{1}{D_j} + L_j(t)}{\mu}$ to guarantee the delay constraint (9). The above analysis can be summarized as Algorithm 1.

Algorithm 1 EMIEP: Choosing the best $m_j^{\text{best}}(t)$ for the Drift-Plus-Penalty Algorithm

- 1: Determine the upper bound of $m_j(t)$ as m_j^{\max}
 - 2: Calculate the minimum of $m_j(t)$: $m_j^{\min}(t) = \frac{\frac{1}{D_j} + L_j(t)}{\mu}$
 - 3: Study the coefficient of $m_j(t)$:
 - 4: **if** $Z_j(t)(T^c + \zeta a_2 - T^{\max}) + V_p(t) a_2 > 0$ **then**
 - 5: **return** $m_j^{\min}(t)$
 - 6: **else**
 - 7: **return** m_j^{\max}
 - 8: **end if**
-

The next step is to test all possible cooling air temperatures to find optimal T^c . This can be done via Algorithm 2.

For a fixed T^c , first call Algorithm 1 and figure out $m_j^{\text{best}}[T^c]$ (line 4) so as to minimize (20) for each application. Plug $m_j^{\text{best}}[T^c]$ into server energy cost Eq. (2) and calculate the total server energy cost.

Substitute $m_j^{\text{best}}[T^c]$ and T^c into the first item of (18) and obtain the simplified drift function. Having computed cooling energy cost depending on Eq. (4), the value of objective function can be obtained by adding server energy cost to cooling energy cost (line 5). For all T^c , the sum of drift function and V multiplied by F^{obj} leads to the value of (20) (line 6). As temperatures change from 10 to 25 by integers, the algorithm chooses $\min T^c$ (line 8), the minimization makes the least values of (20) for all T^c . Once $\min T^c$ is determined, it is convenient to acquire the best number of servers, server energy cost and cooling energy cost (line 11 and 12), and the value of objective function at this time slot t (line 10). Finally, Eq. (13) iterates and updates the virtual queue $\mathbf{Z}(t)$. (line13).

Algorithm 2 EMIEP: Minimizing the Drift-Plus-Penalty Algorithm

- 1: Define:
 - interval: Set of decision epoch
 - Temperature: Set of possible cooling air temperature.
 - Applications: Set of applications.
 - F^{obj} : The value of object function
 - $DPP[T^c]$: The value of Lyapunov Drift-plus-Penalty at T^c
 - 2: **for all** $t \in \text{interval}$ **do**
 - 3: **for all** $T^c \in \text{temperature}$ **do**
 - 4: Call Alg. 1, acquire $m_j^{\text{best}}(T^c)$ **for all** j
 - 5: $F^{\text{obj}} \leftarrow \sum_{j=1}^J p(T^c)(a_1 L_j(T^c) + a_2 m_j^{\text{best}}(T^c)) + p(T^c) * c * f * \rho(T^{SP} - T^c)$
 - 6: $DPP[T^c] \leftarrow \sum_{j=1}^J Z_j(T^c) \{m_j^{\text{best}}(T^c)T^c + \sigma(a_1 L_j(T^c) + a_2 m_j^{\text{best}}(T^c)) - T^{\max} m_j^{\text{best}}(T^c)\} + V \times F^{\text{obj}}$
 - 7: **end for**
 - 8: $\min T^c \leftarrow \arg\min_{T^c} DPP[T^c]$
 - 9: $m^{\text{best}}(t) \leftarrow m[\min T^c]$
 - 10: $\min F^{\text{obj}}(t) \leftarrow F^{\text{obj}}[\min T^c]$
 - 11: $\min \text{Serverprice}(t) \leftarrow \text{serverprice}[\min T^c]$
 - 12: $\min \text{Coolingprice}(t) \leftarrow \text{coolingprice}[\min T^c]$
 - 13: $Z_j(t+1) \leftarrow \max\{Z_j(t) + m^{\text{best}}(t) * T^c(t) + \zeta(a_1 * L_j(t) + a_2 * m^{\text{best}}(t)) - T^{\max} * m^{\text{best}}(t), 0\}$
 - 14: **end for**
-

4 Performance analysis

This section first shows that the minimum time-averaged IDC energy cost can be achieved using a randomized stationary control policy independent of the virtual queue state $\mathbf{Z}(t)$. Then, a performance bound for objective function is derived.

Define $E\{p(t)(E^\pi(t) + C^\pi(t))\}$ as the expected energy cost in the interval $[1, \dots, T]$ under control policy π , and e^* as the minimum achievable $E\{p(t)(E^\pi(t) + C^\pi(t))\}$ over all possible π . If problem Eq. (8) is feasible, then for any $\delta > 0$, there is a policy π^* which depends only on workload and power price which satisfies

$$E\{p(t)(E^{\pi^*}(t) + C^{\pi^*}(t))\} \leq e^* + \delta$$

This result is a direct application of Theorem 4.5 in Ref. [9], which discovers that there is a queue-independent randomized stationary yielding an energy cost arbitrary close to the optimum as long as the problem is feasible. Next the performance bound is given.

Theorem 1 Suppose that $E\{L(Z(0))\} < \infty$. The following results yield:

1) The achievable total energy cost of the EMIEP algorithm can be bounded by

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T E\{p(t)(E(t) + C(t))\} \leq \frac{B}{V} + e^* \quad (21)$$

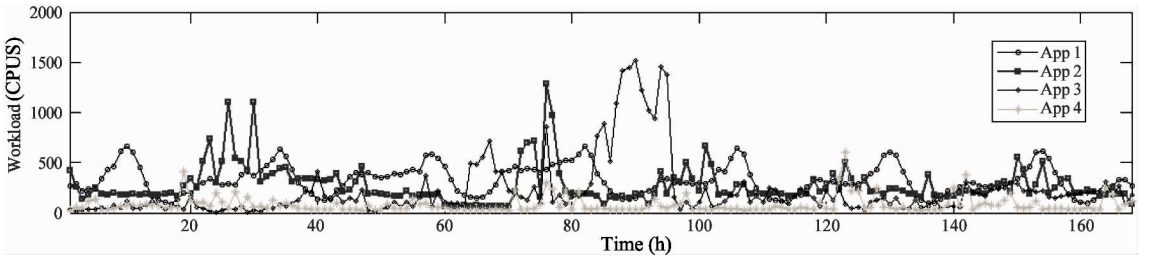


Fig. 2 Workload trace involved in server power

service rate of single server μ is normalized as 1 across all applications. If the minimal and maximal power consumption of a single server is 40W and 80W, parameters in Eq. (1) can be set to $a_1 = 40$ and $a_2 = 40$. Suppose that heat exchange rate is $\zeta = 0.625 \text{K} \cdot \text{s/J}$, the air flow rate is $f = 5 \text{m}^3/\text{s}$, and the maximum allowable server temperature is $T^{\max} = 60^\circ\text{C}$.

5.2 Result analysis

Based on the above set parameters, the performance of the EMIEP algorithm is investigated and compared with a greedy policy, which is obtained by solving the following problem:

2) The virtual queue $\mathbf{Z}(t)$ is mean rate stable, i. e. ,

$$\limsup_{T \rightarrow \infty} \frac{E\{Z_j(t)\}}{T} = 0 \quad (22)$$

According to the above theorems, the article concludes the $[O(1/V), O(V)]$ tradeoff, i. e. , the energy cost can be pushed arbitrarily closed to the optimum as $V \rightarrow \infty$.

5 Numerical evaluations

This section conducts extensive simulations to evaluate the proposed EMIEP algorithm.

5.1 System configuration

All data used in research were accumulated from four applications of Ordos Uni-Cloud Co. , Ltd. , after the authors had spent a week on their Internet Data Center (IDC). Because electricity in EMIEP relates to two factors: server power expenditure and cooling system, workload trace involved in server power contains mean request arrival rate shown in Fig. 2 for interactive web service at intervals per hour.

To analyze energy cost consumed in applications, machine room T^{sp} is set to 25°C . In the light of Ref. [9], the algorithm acquires the heat capacity and the density of the air as $c = 1005 \text{J/kg} \cdot \text{K}$ and $\rho = 1.205 \text{kg/m}^3$ at 25°C . Without loss of generality, the

$$\min PS(t) + PC(t) \quad (23)$$

subject to:

$$\frac{1}{D_j} + L_j(t) - \frac{\mu}{\mu} - m_j(t) \leq 0, \quad \forall j, t$$

$$T^c(t) + \zeta \left(a_1 \frac{L_j(t)}{m_j(t)} + a_2 \right) \leq T^{\max}, \quad \forall j$$

Note that problem (23) greedily minimizes power cost in current slot rather than the long term average power cost.

5.2.1 Total energy cost and delay

Fig. 3 shows that dynamic programming price (DPP) is also a random variable. At the initial state,

its number of samples is not enough that the expectation of objective function is over-estimated, but as time goes by, more samples will be obtained, which leads to more accurate estimation.

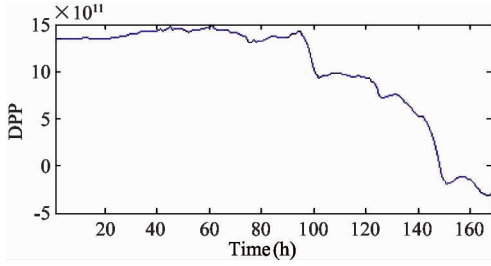


Fig. 3 Value of drift-plus-penalty objective function with $V = 10$

Different vertical ordinates shown in Fig.4(a), 4(b) and 4(c) individually denote total, server and cooling energy costs V , the horizontal ordinate, controls the weight of Lyapunov drift and objective function. The experiment also plots the performance of greedy strategy in the figure, as shown in Fig.4(a) and 4(b), the total and server energy costs descend in steps with the increase of V . In this way, the objective function is efficiently optimized. Though the cooling energy cost mildly increases at $V = 10^8$, $V = 10^9$ and $V = 10^{10}$ shown in Fig.4(c), this situation has not affected the general tendency of energy cost as V grows. When V grows to $V = 10^{11}$, this EMIEP outperforms greedy strategy.

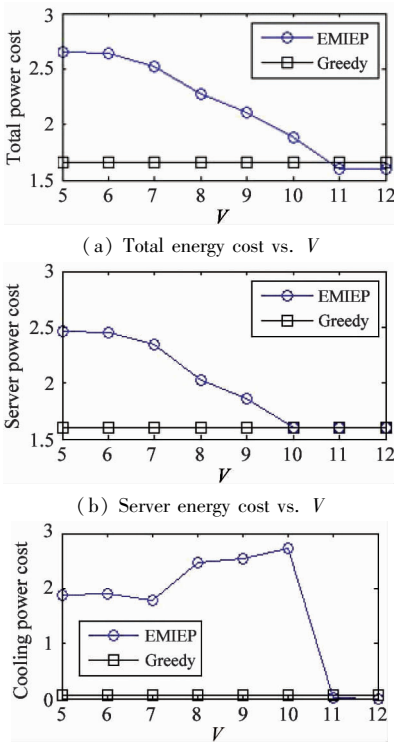


Fig. 4 Energy cost vs V

The QoS (delay) for each application in Fig. 7 discovers that the 10ms QoS requirement is met in this EMIEP strategy. Fig. 5 and Fig. 6 show the number of servers allocated to each application with time and the variation of drift-plus-penalty objective function, respectively. It can be shown that this EMIEP can effectively minimize the system power cost while ensuring QoS requirement.

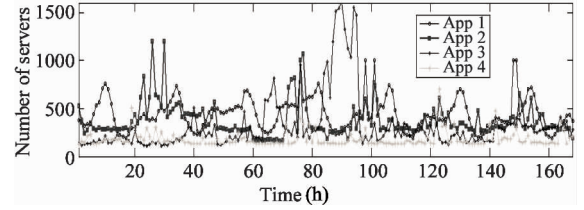


Fig. 5 Servers allocated to each application with $V = 10$

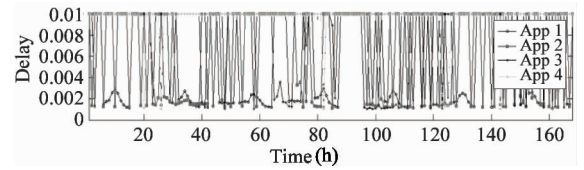


Fig. 6 QoS (delay)

5.2.2 Cooling temperature and server temperature

Compared with Fig.4(c), the variation trend of cooling cost in Fig. 7 goes to different way. The case explains the fact that the lower the temperature of cooling air is set, the higher the cooling power consumption and cost.

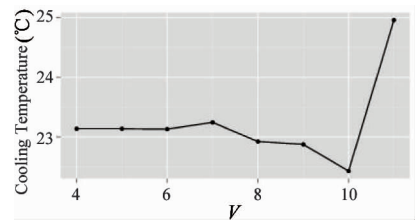


Fig. 7 Cooling air temperature vs. V

Since energy consumption directly influences server temperature, the server temperatures as shown in Fig. 8 from four applications constantly keep heading

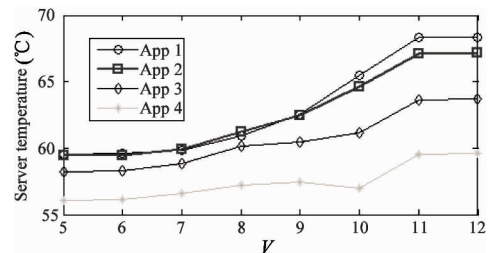


Fig. 8 Server temperature vs. V

up but occasionally fluctuate at $V = 10^9$ and $V = 10^{10}$ in application 3 and 4. Raising V to 10^{11} slightly increasing the average temperature, significant power cost savings can be obtained.

6 Conclusion

This work concentrates on greening the data center by minimizing time-averaged expected energy cost subject to QoS and average temperature constraints.

Since the workload distribution cannot be obtained in advance, it is challenging to design a dynamic control algorithm to achieve this goal.

To address this opinion, the algorithm leverage the Lyapunov Optimization technique and develop an algorithm to approximately solve the Expense Minimization of IDC's Electricity Power problem. By evaluating the simulation experiment, the algorithm can be pushed arbitrarily close to optimal solution as control parameter V is raised. Based on real workload trace from Ordos Uni-Cloud Technology Co., Ltd. to simulate the proposed algorithm, numerical results illustrate that the algorithm can practically reduce total energy cost while guaranteeing QoS and temperature constraints.

References

- [1] Qureshi A, Weber R, Balakrishnan H, et al. Cutting the electric bill for internet-scale systems. *ACM SIGCOMM Computer Communication Review*, 2009, 1:123-134
- [2] Dou H, Qi Y, Wang P J, et al. Hybrid power control and electricity cost management for distributed internet data centers in cloud computing. In: Proceedings of the 10th International Conference on Web Information System and Application, Yangzhou, China, 2013. 394-399
- [3] Yao J G, Liu X, He W B, et al. Dynamic control of electricity cost with power demand smoothing and peak shaving for distributed internet data centers. In: Proceedings of the

- 2012 IEEE 32nd International Conference on Distributed Computing Systems, Macau, China, 2012, 67: 416-424
- [4] Wang C, Urgaonkar B, Wang Q, et al. A hierarchical demand response framework for data center power cost optimization under real-world electricity pricing. In: Proceedings of the 2014 IEEE 22nd International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Paris, France, 2014, 45: 305-314
- [5] Xin Z, Reda S. Power budgeting techniques for data centers. *IEEE Transactions on Computers*, 2015, 64 (8): 2267-2278
- [6] Zhang Y W, Wang Y, Wang X. TEStore: exploiting thermal and energy storage to cut the electricity bill for data-center cooling. In: Proceedings of the 8th International Conference on Network and Service Management International Federation for Information Processing, Las Vegas, USA, 2012, 1:19-27
- [7] Kaushik R T, Nahrstedt K. A data-centric cooling energy costs reduction approach for big data analytics cloud. In: Proceedings of the 2012 International Conference on High Performance Computing, Networking, Storage and Analysis, Madrid, Spain, 2012, 1: 1-11
- [8] Li S, Le H, Pham N, et al. Joint optimization of computing and cooling energy: analytic model and a machine room case study. In: Proceedings of the International Conference on Distributed Computing Systems, Macau, China, 2012, 1: 396-405
- [9] Neely M J. Stochastic network optimization with application to communication and queueing systems. *Morgan & Claypool*, 2010, 4:53-62

Zhang Ran, is a graduate pursuing Ph.D in School of Mathematics and Statistics of Central South University. He received B.S. in applied mathematics from Branch Campus of Peking University in 1994 and M.S in computer science from Xidian University in 2007. His research interests focus on probability and applied statistics, machine learning, IDC resource management, and data mining.