Dynamic Server Provisioning for Energy Efficient Large Scale Video-on-Demand Systems

Jian Yang, Ke Zeng

School of Information Science and Technology, University of Science and Technology of China (USTC), Hefei, Anhui 230027, China, e-mail: jianyang@ustc.edu.cn, zknkcq@mail.ustc.edu.cn

Abstract: In this paper, we propose a server provisioning strategy for energy conservation in large scale VOD systems, which dynamically turns on/off servers in order to adaptively tailor active servers to dynamic user load. By defining quality-of-service (QoS) in terms of system overload probability, we reinterpret the energy conservation problem into minimizing the number of active servers subject to a QoS requirement. Relying on designing a recursive least square (RLS) based online user number predictor and constructing a large deviation based overload probability estimation model, we derive a practical dynamic server provisioning strategy which does not require any prior knowledge about the future workload. Finally, the experiments are carried out based on synthetic and real workload respectively to investigate the achievable performance of the proposed strategy.

Keywords: Video-on-demand, server energy conservation, quality-of-service, load migration, request dispatching.

1. Introduction

Streaming media is undergoing a dramatic growth fuelled by a variety of applications such as internet protocol television (IPTV) and Telco video. Naturally, video-on-demand (VOD) service has to face a challenge of the tremendous concurrent users increase. In order to satisfy the increasingly demanding traffic and the growing consumer population, most VOD systems are deployed in a form of massive server clusters for providing low cost, high performance, availability and scalability. However, the servers today consume ten times more power than they did ten years ago [1], which implies that the large scale server clusters may result in a high power consumption. In this paper, we consider the problem of energy conservation in large scale VOD server clusters.

In recent years, the problem of energy conservation for server clusters has received an increasing attention from both academia and industry. Bianchini *et al.* [2] gave an overview of different power and energy management techniques of server systems. Gandhi *et al.* [3] considered the problem of allocating an available power budget among servers to minimize mean response time. Chen *et al.* [4] presented a solution to the problem of reducing server energy cost at hosting center running multiple applications towards the goal of meeting performance based on Service Level Agreements (SLA). Qureshi *et al.* [5] characterized the variation due to fluctuating electricity prices and stated that existing distributed systems should be able to exploit this variation for significant economic gains. Chase *et al.* [6] studied dynamic provisioning for web cluster to improve the energy efficiency. Chen *et al.* [7] proposed power saving techniques for connection-intensive services, and evaluated the techniques by using data traces from Windows Live Messenger. Other works [8-11] also studied dynamic provisioning and load dispatching algorithms for web clusters or data centers to improve the energy efficiency.

It should be noted that the most prior works focus on generic requests of web services, rather than multimedia jobs. To the best of our knowledge, energy saving in large scale VOD systems has not been covered yet. Our works for power-efficient large scale VOD server clusters are motivated by two observations. First, the number of users varies largely during a day [13], which is based on a large VOD system deployed by China Telecom. Second, as shown in [3, 7], an idle server may consume around 60% of the peak power, because the power required to run the OS and hardware is not ignorable. These observations imply that we can achieve substantial energy saving by adjusting the active server according to the time-varying workload, especially shutting down idle servers during off-peak hours.

In this paper, we design a recursive least squares (RLS) based predictor to estimate the forthcoming number of users corresponding to each video, which assists our strategy to determine the server provisioning for the near instantaneous workload without any prior statistical knowledge. In order to conceive a QoS provisioning energy conservation strategy for large scale VOD systems, we define the

QoS requirement in terms of the overload probability. Then, we apply large deviation theory [14] to estimate the overload probability that the bandwidth provided by the current active servers cannot satisfy the required bandwidth for the predicted workload. Finally, we use the overload probability to derive an adaptive strategy to find the minimum cluster scale satisfying the QoS requirement.

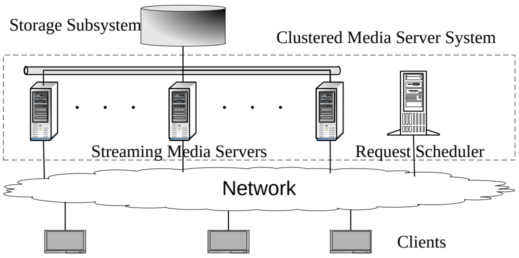


Figure 1. The clustered VOD systems

2. System Model

Consider a clustered VOD system as shown in Figure 1, which consists of a request scheduler, a collection of media servers and a storage subsystem. The request scheduler catches the request and distributes it to the chosen streaming media server. The streaming media server delivers the video stream to the requested client through the network. The storage subsystem provides a large storage capacity for storing video objects and high data throughput for supporting multiple concurrent retrievals of video objects. Suppose that all the videos are variable bit-rate (VBR).

Consider at time t, there are N users and k active servers in a VOD system. Suppose all the servers have the same bandwidth c Mbps. Then, the aggregated bandwidth capacity is denoted as $C = kc$ (Mbps). Let J denote the number of videos. The bandwidth required by the user i watching a video at time t is denoted by $X_i(t)$. Hence, we define the overload probability as

$$P_{overload} = P(\sum_{i=1}^{N} X_i(t) > C) \tag{1}$$

Then, the energy conservation problem for large VOD server clusters with QoS provisioning can be formulated as

$$\begin{aligned} &\min_k \quad C = kc \\ &s.t. \quad P(\sum_{i=1}^{N} X_i(t) > C) < \varepsilon \end{aligned} \tag{2}$$

where ε is a desired QoS requirement in terms of the overload probability. Obviously, estimating the overload probability is a key step to solve the problem (2). In this paper, a practical estimation method will be proposed to predict the overload probability in (2) by using two steps: the first step is to design a RLS based predictor for the user number N , followed by the second step that applies the large deviation principle to estimate the overload probability.

3. Dynamic Server Provisioning for Large Scale VOD Clusters

In this section, we will derive a prediction based dynamic server provisioning strategy for large scale VOD clusters. Firstly, a RLS predictor is designed for predicting the forthcoming user number for each video. Then, based on the user number estimate, we apply the large deviation principle to calculate the overload probability. Finally, a dynamic server provision strategy is presented to solving the problem (2) by using the above two steps.

3.1 RLS based predictor for user number

The VOD server cluster may waste a substantial amount of energy due to the low percentage of the peak load time. Therefore, if we can use the workload variation to dynamically adjust the scale of the server cluster, significant energy conservation may be achieved. However, there is no prior knowledge available about the user number variation. Additionally, the characteristics of VOD service are significantly different from connection-intensive service like MSN, because the video traffics are heterogeneous and their popularity will change upon time. To overcome the above difficulties, we will propose a measurement based online predictor for the user number of each video, which assists the proposed strategy to activate early adjustment of the active server scale to compensate the initialization delay of starting a physical server.

Since Recursive Least Squares (RLS) estimation method [16] has fast convergence and tracking capability in non-stationary environment, we adopt RLS technique to design a measurement-based online predictor for the user number of each video. Let $n_j(l)(j=1,2,\cdots,J)$ denote the user number corresponding to the j th video at time l. Then, define $\mathbf{w}_j=[w_j(0),w_j(1),\cdots,w_j(p-1)]^T$ as the predictor's weight vector corresponding to the j-th video. We apply a 1-step p th-order predictor to estimate the user number as

$$\hat{n}_j(l+1)=\sum_{t=0}^{p-1}w_j(t)n_j(l-t) \quad (3)$$

Then, the prediction error is given as

$$e_j(l+1)=n_j(l+1)-\hat{n}_j(l+1) \quad (4)$$

In order to find the weight vector, we consider the following sum of exponentially weighted error squares

$$\sum_{i=0}^{l}\lambda^{l-i}e^2(i) \quad (5)$$

where the forgetting factor λ is between 0 and 1. Choosing $\lambda<1$ is especially useful for tracking nonstationary changes. Applying the RLS technique to minimize the criterion (5), we can derive the weight vector updating rule as follows:

$$\begin{aligned}\mathbf{k}_j(l)&=\frac{\lambda^{-1}\mathbf{P}_j(l-1)\mathbf{n}_j(l)}{1+\lambda^{-1}\mathbf{n}_j^T(l)\mathbf{P}_j(l-1)\mathbf{n}_j(l)}\\ \mathbf{w}_j(l)&=\mathbf{w}_j(l-1)+\mathbf{k}_j(l)e(l)\\ \mathbf{P}_j(l)&=\lambda^{-1}\mathbf{P}_j(l-1)-\lambda^{-1}\mathbf{k}_j(l)\mathbf{n}_j^T(l)\mathbf{P}_j(l-1)\end{aligned} \quad (6)$$

where $\mathbf{n}_j(l)=[n_j(l),n_j(l-1),\cdots,n_j(l-p+1)]^T$ and $\mathbf{w}_j(l)$ is the estimate of $\mathbf{w}_j$ at time l. The simplest way to choose the initial value is to set $\mathbf{P}_j(0)=\mu\mathbf{I}$.

At time l, we can predict the user number of video j at time $l+1$ as

$$\hat{n}(l+1)=\mathbf{w}_j^T(l)\mathbf{n}_j(l) \quad (7)$$

In order to provide QoS, our strategy aims to provide enough bandwidth capacity for the maximum number of users at time $l+1$. However, the user number $n_j(l+1)$ predicted by (7) may not be the maximum user number of the video j due to the prediction error. Therefore, we extend the predictor (7) as

$$n'_j(l+1)=\hat{n}_j(l+1)+\alpha\max\{\max_{i\in[l-\tau,l]}\{e(i)\},0\} \quad (8)$$

where the scale factor α satisfies $\alpha\geq 1$ and $\tau>0$. The second item in the right side of (8) is to compensate the prediction error of (7) by scaling the maximum value of the recent τ prediction errors, in order to prevent the QoS degradation due to underestimating the user number. The scale factor α should be carefully chosen, because excessive α may result in overestimation which leads to over-provisioning of servers and power wasting, while small α may result in underestimation, which leads to under-provisioning and QoS degradation.

3.2 Overload probability estimation

The prediction technique in the above section assists us to estimate the forthcoming user number of each video. However, the user number is not sufficient to characterize the workload in VOD service due to the heterogeneous traffic of each video. Hence, we will further apply large deviation theory [14, 15] to develop an overload probability model by using the predicted user number.

Let n_j denote the predicted number of the forthcoming user corresponding to the video j $(j=1,2,\cdots,J)$. We further define the bandwidth demand of the i th user watching video j as X_{ji}. Then, the aggregated bandwidth S required by the forthcoming users is formulated as

$$S=\sum_{i=1}^{N}X_i(t)=\sum_{j=1}^{J}\sum_{i=1}^{n_j}X_{ji} \quad (9)$$

Then, the overload probability can be rewritten as

$$P_{overload}=P(S>C) \quad (10)$$

where $C=kc$ is the total bandwidth provided by k active servers.

Since X_{ji} $(i=1,2,\cdots,n_j)$ are corresponding to the same video j, the random variables X_{ji} are *i.i.d*, and they have the same logarithmic moment generating functions

$$M_j(\theta)=\log E[e^{\theta X_{ji}}] \quad (11)$$

By applying the Chernoff bound in the context of the large deviations approximation [14, 15] to $P(S>C)$, we obtain

$$\log P(S>C)\approx\inf_{\theta}[\sum_{j=1}^{J}n_jM_j(\theta)-\theta C] \quad (12)$$

Hence, the overload probability can be estimated as follows:

$$P_{overload}=\exp\{\inf_{\theta}[\sum_{j=1}^{J}n_jM_j(\theta)-\theta C]\} \quad (13)$$

With the above estimation equation, we can determine whether the current bandwidth $C=kc$ provided by k active servers satisfies the QoS requirement, ε , for the forthcoming workload by using the criterion

$$\exp\{\inf_{\theta}[\sum_{j=1}^{J}n_jM_j(\theta)-\theta C]\}<\varepsilon \quad (14)$$

As shown in the next section, we will rely on this criterion to solve the problem (2) to find the minimum active server number.

Below we will further give the remarks on the computation of the overload probability. To calculate the overload probability, the explicit form for $M_j(\theta)$ is required, which means that we have to estimate the bandwidth distribution of video j . We define the possible bandwidth value set for the video j at different time instants as $B_j=\{b_{j1},b_{j2},\cdots,b_{jm}\}$ where m is the number of possible bandwidth values. Then, we may sample the demanded bandwidth with the period T_s, say a GOP time duration. Let D_j and d_{jk} respectively denote the total sample number and the number that the sampled bandwidth is b_{jk} $(k=1,2,\cdots m)$. Hence, the distribution $\pi_j(k)=P(X_{ji}=b_{jk})$ can be estimated according to $\pi_j(k)=d_{jk}/D_j$. Then, the explicit form of $M_j(\theta)$ in (11) can be rewritten as

$$M_j(\theta)=\log\{\pi_j(k)e^{\theta b_{jk}}\} \quad (15)$$

Then, based on the (13) we can estimate the overload probability via computers.

3.3 Prediction-based dynamic server provisioning strategy

In this section, we will apply our proposed prediction technique and overload probability model to find the solution of the problem (2).

Suppose at every beginning of the scheduling interval, say time l , there are $n_j(l)$ users watching the j th video $(j=1,2,\cdots,J)$ in a VOD system. Let $K(l)$ denote the active servers at interval l . Our aim is to predict the active server number at interval $l+1$ (i.e., $K(l+1)$), which provides the QoS assurance for the forthcoming workload. To achieve this, we first use the predictor (8) to estimate the user number $n'_j(l+1)$ $(j=1,2,\cdots,J)$ during next scheduling interval.

Then the number of active servers is determined as follows. The proposed strategy applies the criterion (14) to check whether the actual number, $K(l)$, of active servers at current time could satisfy the QoS parameter. Let k denote the temporary variable for active server number. Its initial value is $K(l)$. Two scenarios may appear according to the criterion (14), and the corresponding strategy is described as:

1) If the criterion (14) is not satisfied, this implies that the aggregated bandwidth provided by k active servers cannot satisfy the desired QoS for the forthcoming workload. Then, the process that we increase the active server number by $k=k+1$ is repeated until the criterion (14) is satisfied. Then, the number of active servers provisioning for the forthcoming workload should be $K(l+1)=k$.

2) Otherwise, the current k active servers can provide bandwidth capacity for accommodating the forthcoming workload with QoS assurance. But it unnecessarily implies that k is the minimum number of active servers that provide the desired QoS for the forthcoming workload. Therefore, we repeat $k=k-1$ until the criterion (14) is not satisfied. The minimum number of active servers for problem (2) is $K(l+1)=k+1$.

Once we determine the active server number for the workload at time $l+1$, i.e., $K(l+1)$, we can turn on or off servers accordingly. The strategy is summarized in Algorithm 1.

Algorithm 1 Server Provisioning Algorithm

```
Beginning of the $l$ th scheduling interval
Input: $K(l), n_j(l), \mathbf{w}_j(l-1) (j=1,\cdots,J)$
Output: $K(l+1)$
for $j=1$ to $J$ do
    Calculate $\mathbf{w}_j(l)$ according to (6);
    Calculate $n'_j(l+1)$ according to (8);
```

```
end for
$k = K(l)$ and $C = kc$;
Calculate $P_{overload}$ according to (13);
if $P_{overload} < \varepsilon$ then
    while ($P_{overload} < \varepsilon$) and $(k > 1)$ do
        $k \leftarrow (k-1)$;
        Update $P_{overload}$ according to (13);
    end while
    $K(l+1) = k + 1$;
else
    while ($P_{overload} > \varepsilon$) do
        $k \leftarrow (k+1)$;
        Update $P_{overload}$ according to (21);
    end while
    $K(l+1) = k$;
end if
```

4. Load Migration

In order to provide a more complete framework for energy conservation in large VOD server cluster systems, we further discuss load migration in the context of power saving, which may affect our proposed dynamic server provisioning strategy.

When tailoring the excessive active servers to the target optimum active server number determined by Algorithm 1, it is ideal that enough idle servers are available to turn off instantly for energy conservation. However, there may be insufficient idle servers due to long-live video sessions and their loose distribution among the active servers, which implies that the proposed strategy has to wait and turn off the server which is becoming idle. Unlike most web services the video sessions may last long time, typically more than an hour. Hence, a non-idle server may wait a long time before it becomes idle even if no requests are distributed to it any more. As a result, it may take a long time to tailor current cluster scale to the optimum one, which may affect the energy conservation efficiency of our proposed strategy. In order to reduce the detrimental effect on energy conservation, we rely on load migration technique to obtain more idle servers by migrating the sessions to other servers with available capacity. The simplest way to implement the load migration for VOD service is to apply client reconnections. Specifically, the candidate server for load migration sends a message to notify its clients that it will be turned off. Any client that receives this message will send a request to an alternative server for the rest video data. Since large concurrent reconnection may increase migration delay, we always select the servers with a small number of video sessions as the candidates for load migration. In order to achieve this, we define a threshold N_t to select the candidate servers. When the proposed strategy requires more idle servers, a server with its session number less than N_t will be chosen to migrate its workload, and it will be shut down as it becomes idle. The effect of N_t on our proposed strategy is investigated in Section V.

5. Experimental Results

In this section, the performance of our proposed dynamic server provisioning strategy is investigated based on a synthetic workload and a real workload.

5.1 Experiment settings

In the simulation, we considered a clustered VOD system with 20 servers. The streaming capacity of each server was set to 600Mbps, which is the streaming bandwidth of a single Kasenna SpeedBase Media Server [20]. The delay of starting a streaming server is assumed to be uniformly distributed in [2,4] minutes. Four video sequences, namely, “Silence of the Lambs”, “Star Wars IV”, “NBC 12 News”, “Tokyo Olympics” [17] were used in the simulation. Each has a duration of 30min and was compressed with a frame rate $F = 30$ frames/s and a GOP size of 16 frames. Further detail of the videos could be found in [17].

Video request arrivals, synthesized based on a user arrival distribution model and extracted from real workload trace, were respectively used to investigate the performance of the proposed strategy. User arrival model can assist us to perform experiments with various request patterns which may characterize different VOD applications. In the simulation, we implemented the modified Poisson distribution model proposed in [13] based on the seven months' VOD logs provided by China Telecom. We further investigated the performance of our proposed strategy in a specific practical application by using YouTube trace [12]. In order to simulate the user behaviour in VOD systems, we used two typical video popularity distributions according to [18, 19]. The distributions are given in Table 1. Type I corresponds to the circumstance where videos have similar popularity, while Type II corresponds to the scenario with different popularity. Time is discretized into time slots, and each slot has a GOP duration, i.e., 16/30 second. Our proposed strategy was invoked once every 4 minutes.

Table 1: Video Access Popularity

Type	**Video 1**	**Video 2**	**Video 3**	**Video 4**
I	0.357	0.257	0.2	0.186
II	0.621	0.205	0.107	0.068

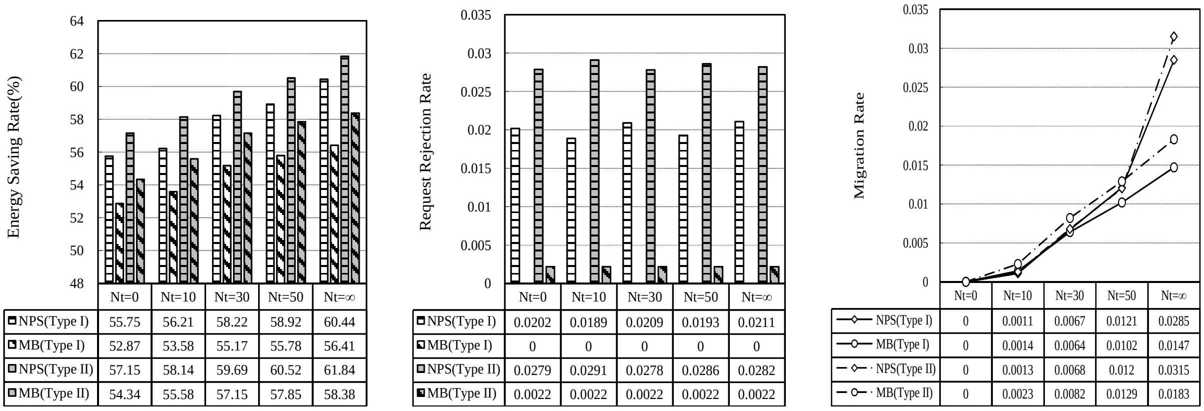
For our proposed strategy, a 5th-order predictor was applied, i.e. $p=5$ in (3). The initial parameters for RLS were set as follows: $\mathbf{w}_j(0)=[1/p,1/p,\cdots,1/p]$, forgetting factor $\lambda=0.998$, $\mathbf{P}_j(0)=10^{-4}\mathbf{I}$ and $\tau=20$ in (16). For the notational convenience, our proposed online measurement based strategy is referred to MB. For comparison, we also implemented none-prediction based scheme (NPS), which turns on or turns off a server when the number of current active servers is not enough or it is larger than the demanded capacity, respectively.

In order to evaluate the performance of our proposed strategy, we define three performance metrics, i.e., Energy Saving Rate, Request Rejection Rate and Migration Rate. The number of independent runs for average simulation results was set to 50.

Simulation Results

We set $\alpha=1.0$ in (8) and $\varepsilon=10^{-6}$ for our proposed strategies. For the synthetic user arrivals, we applied the modified poisson model with $\lambda=17$ and by varying maximum user arrivals N in [13] in order to generate the typical daily access pattern of the commercial television industry: dropping gradually during the early morning (12AM-7AM) and the afternoon (2PM-5PM), while climbing up to a peak in noon break (Noon-2PM) or after work (6PM-9PM). Since two video popularities, i.e., Type I and Type II in Table 1 were applied, we could produce two corresponding video access workload. With these workloads, we performed the simulations respectively for different migration thresholds, i.e. $N_t=0,10,30,50,\infty$. $N_t=0$ means that no load migration scheme was applied in the VOD cluster, while $N_t=\infty$ implies that the workload migration would be always invoked. The simulation results in terms of Energy Saving Rate, Request Rejection Rate and Migration Rate were plotted in **Figure** 2. Similarly, we also utilized the user arrivals extracted from YouTube trace and the video access popularity Type I and II to generate workloads for simulations. The corresponding simulation results were given in **Figure** 3.

From **Figure** 2(a) and **Figure** 3(a), we can observe that for any given N_t, applying the proposed MB strategy and NPS both largely reduce the energy cost of the VOD system, and Energy Saving Rate corresponding to NPS is about 3.5% higher than that of our proposed strategy, which means NPS achieves somewhat better energy saving performance. However, the simulation results in **Figure** 2(b) and **Figure** 3(b) show that Request Rejection Rate of MB strategy is lower than 10 percentage of NPS's, which implies that compared to NPS our proposed MB strategy substantially improves QoE at a cost of

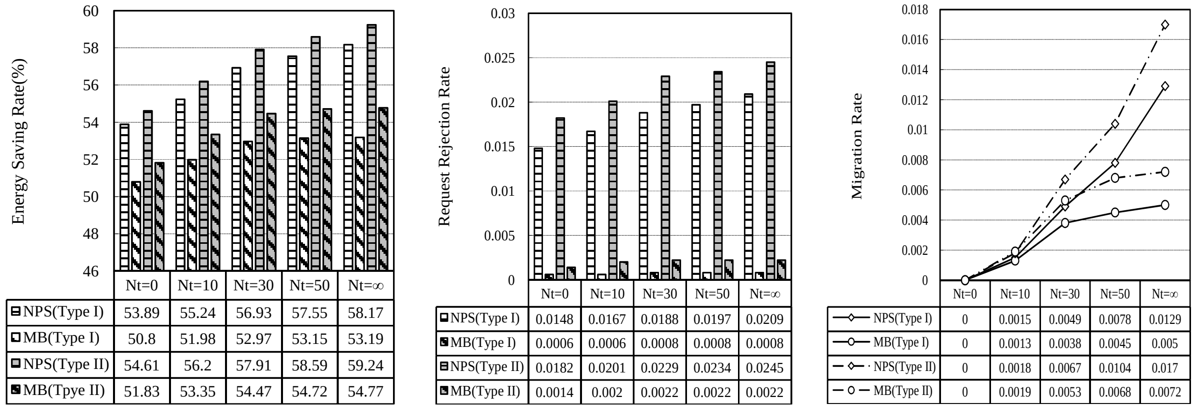


	Nt=0	Nt=10	Nt=30	Nt=50	Nt=∞
▤NPS(Type I)	55.75	56.21	58.22	58.92	60.44
▨MB(Type I)	52.87	53.58	55.17	55.78	56.41
▤NPS(Type II)	57.15	58.14	59.69	60.52	61.84
▨MB(Type II)	54.34	55.58	57.15	57.85	58.38

	Nt=0	Nt=10	Nt=30	Nt=50	Nt=∞
▤NPS(Type I)	0.0202	0.0189	0.0209	0.0193	0.0211
▨MB(Type I)	0	0	0	0	0
▤NPS(Type II)	0.0279	0.0291	0.0278	0.0286	0.0282
▨MB(Type II)	0.0022	0.0022	0.0022	0.0022	0.0022

	Nt=0	Nt=10	Nt=30	Nt=50	Nt=∞
—◇— NPS(Type I)	0	0.0011	0.0067	0.0121	0.0285
—○— MB(Type I)	0	0.0014	0.0064	0.0102	0.0147
– ◇– · NPS(Type II)	0	0.0013	0.0068	0.012	0.0315
– ○– · MB(Type II)	0	0.0023	0.0082	0.0129	0.0183

Figure 2. Simulation results for NPS and MB in the scenario of synthetic workload.



	Nt=0	Nt=10	Nt=30	Nt=50	Nt=∞
▤NPS(Type I)	53.89	55.24	56.93	57.55	58.17
▨MB(Type I)	50.8	51.98	52.97	53.15	53.19
▤NPS(Type II)	54.61	56.2	57.91	58.59	59.24
▨MB(Tpye II)	51.83	53.35	54.47	54.72	54.77

	Nt=0	Nt=10	Nt=30	Nt=50	Nt=∞
▤NPS(Type I)	0.0148	0.0167	0.0188	0.0197	0.0209
▨MB(Type I)	0.0006	0.0006	0.0008	0.0008	0.0008
▤NPS(Type II)	0.0182	0.0201	0.0229	0.0234	0.0245
▨MB(Type II)	0.0014	0.002	0.0022	0.0022	0.0022

	Nt=0	Nt=10	Nt=30	Nt=50	Nt=∞
—◇— NPS(Type I)	0	0.0015	0.0049	0.0078	0.0129
—○— MB(Type I)	0	0.0013	0.0038	0.0045	0.005
— ◇— · NPS(Type II)	0	0.0018	0.0067	0.0104	0.017
— ○ — MB(Type II)	0	0.0019	0.0053	0.0068	0.0072

Figure 3. Simulation results for NPS and MB in the scenario of real workload.

increasing a minor energy consumption.

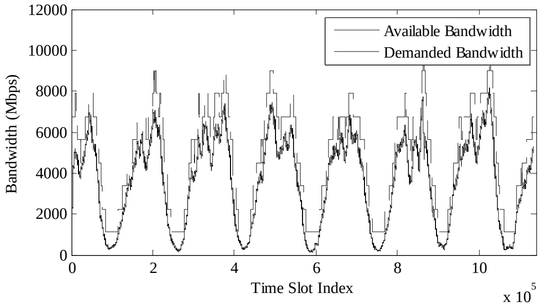


Figure 4. Demanded bandwidth versus Available Bandwidth Provided by active servers

In **Figure** 4, we presented the process of dynamic server provisioning based on MB for real workload. It is shown that our prediction-based MB strategy adaptively reserves a small extra capacity for compensating the future increased workload. Naturally, it may result in more energy cost, but can substantially reduce Request Rejection Rate. While NPS just use the current workload information to provide the active servers, which cannot guarantee enough capacity for the future workload requirement due to lacking prediction capability. Hence, our proposed MB strategy provides better trade-off between the energy cost and QoE.

Figure 2 (a) and **Figure** 3 (a) also show that increasing the migration threshold N_t can reduce the energy consumption, which confirms the statement that load migration can assist us to further reduce the energy cost. However, **Figure** 2(c) and **Figure** 3(c) show that Migration Rate may increase as N_t increases, which may impose more system cost to migrate the video sessions. Moreover, **Figure** 2(b) and **Figure** 3(b) demonstrate that the load migration for different N_t has little effect on Request Rejection Rate. In fact, the load migration is applied to redistribute the workload on the active servers so that the strategy can find enough idle servers to power off, whose effect on Request Rejection Rate should be tiny. Therefore, we can select appropriate N_t to make a good trade-off between the energy saving and the migration-induced cost without considering its effect on Request Rejection Rate.

6. Conclusions

In this paper, the energy conservation problem in large scale VOD systems was studied. We proposed an explicit model for dynamic server provisioning, which minimizes the number of active servers subject to the constraint of QoS requirement in terms of overload probability. A RLS based user number predictor and a large deviation principle based overload probability estimation model were proposed in order to predict the future system overload probability. The proposed strategy utilizes the overload probability estimate to seek the minimum active server number for QoS provisioning. The experiments were performed with synthetic and real workload respectively to investigate the achievable performance of the proposed strategy.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No. 61174062), the Fundamental Research Funds for the Central Universities as well as Program for New Century Excellent Talents in University (NCET-08-0522).

REFERENCES

1. U.S. Environmental Protection Agency, **Server and Data Center Energy**, 2007.
2. BIANCHINI, R., RAJAMONY, R., **Power and Energy Management for Server Systems**, IEEE Computer, vol. 37, 2004, pp. 69-76.
3. GANDHI, A., HARCHOL-BALTER, M., et al., **Optimal Power Allocation in Server Farms**, Proc. ACM SIGMETRICS/ Performance, 2009.
4. CHEN, Y., DAS, A., et al., **Managing Server Energy and Operational Costs in Hosting Centers**, Proc. ACM SIGMETRICS, 2005.
5. QURESHI, A., WEBER, R., *et al.*, **Cutting the Electric Bill for Internet-Scale Systems**, Proc. ACM SIGCOMM, 2009.
6. CHASE, J.S., ANDERSON, D.C., et al., **Managing Energy and Server Resources in Hosting Centers**, Proc. ACM SOSP, 2001.
7. CHEN, G., HE, W., et al., **Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services**, Proc. USENIXNSDI, 2008.
8. JOUKOV, N., SIPEK, J., **GreenFS: Making Enterprise Computers Greener by Protecting Them Better**, in Proc. ACM EuroSys, 2008.
9. PADALA, P., HOU, K., et. al., **Automated Control of Multiple Virtualized Resources**, Proc. ACM Eurosys, 2009.
10. GANDHI, A., GUPTA, V., et al., **Optimality Analysis of Energy-Performance Trade-offfor Server Farm Management**, Proc. IFIP Performance, 2010.
11. URGAONKAR, R., KOZAT, U.C., et al., **Dynamic Resource Allocation and Power Management in Virtualized Data Centers**, Proc. IEEE/IFIPNOMS, 2010.
12. http://traces.cs.umass.edu/index.php/network /Network.
13. YU, H., ZHENG, D., et al., **Understanding User Behavior in Large-Scale Video-on-Demand Systems**, Proc. ACM EuroSys, 2006.
14. BILLINGSLEY, P., **Probability and Measure**, 2nd ed., Wiley, NewYork, 1986.
15. KELLY, F.P., **Effective bandwidths at multi-class queues**, Queueing Systems: Theory and Applications, vol.9, 2001, pp.5-16.
16. HAYKIN, S., **Adaptive Filter Theory Third Edition**. Upper Saddle River, NJ: Prentice-Hall, 1996.
17. http://trace.eas.asu.edu/mpeg4/index.html.
18. ALMEIDA, J.M., KRUEGER, J., et al., **Analysis of educationa lmedia server workloads**, Proc. ACM NOSSDAV, 2001.
19. CHERKASOVA, L., GUPTA, M., **Analysis of Enterprise Media Server Workloads: Access Patterns, Locality, Content Evolution, and Rates of Change**, IEEE/ACM Trans. Netw., vol. 12, 2004, pp. 781-794.
20. http://www.kasenna.com/kasenna/static/products/hardware.html.