Server Frequency Control Using Markov Decision Processes

Lydia Y. Chen
IBM Zurich Research Laboratory
Email: yic@zurich.ibm.com

Natarajan Gautam
Texas A&M University
Email: gautam@tamu.edu

***Abstract*—For a wide range of devices and servers, Dynamic Frequency Scaling (DFS) can reduce energy consumption to various degrees by appropriately trading-off system performance. Efficient DFS policies are able to adjust server frequencies by extrapolating the transition of the highly varying workload without incurring much of implementation overhead. This paper models DFS policies of a single server using Markov Decision Processes (MDP). To accommodate the highly varying nature of workload in the proposed MDP, we adopt fluid approximation based on continuous time Markov chain and discrete time Markov chain modeling for the fluid workload generator respectively. Accordingly, we design two frequency controllers (FC), namely C-FC and D-FC, corresponding to the continuous and discrete modeling of the workload generator. We evaluate the proposed policies on synthetic and web traces. The proposed C-FC and D-FC schemes ensure performance satisfaction with moderate energy saving as well as ease of implementation, in comparison with existing DFS policies.**

I. Introduction

Various power management mechanisms, such as turning devices and servers on/off, dynamic frequency scaling (DFS) and workload consolidation, have been proposed to maintain an efficient power to performance ratio for mobile gadgets and large data centers. DFS technology is commonly embedded in modern devices owing to its lower implementation overhead. Moreover, DFS can be flexibly integrated with the afore-mentioned power-performance management mechanisms. Most existing DFS mechanisms [9], [5] reactively modulate the frequency if the performances metrics being monitored, such as processor utilization, response time etc, are observed to veer from the target values. As there is often no information on future workload, reactive DFS schemes can be rather greedy and even aggravate the performance degradation, especially in the presence of highly varying workloads, although reasonable energy savings might be still achieved. The workloads of many modern devices [14] are found to be highly varying and difficult to characterize and predict because of the long-range dependency among arriving requests. It is not a trivial endeavor to design a proactive optimal DFS scheme that modulates the frequency based on the system state monitored, the workload prediction, and performance.

Markov Decision Processes (MDP) is a common methodology to develop optimal decision rules based on the observation and prediction of the system state. Many single-server control problems, such as determining the optimal admission and service rates [2] to minimize response time, are resolved by employing MDP modeling because of its optimality in achieving system performance objectives. Thus, MDP seems a natural choice for designing a proactive DFS scheme that requires workload characterization, low on-line implementation overhead, and stringent performance requirements. To apply MDP modeling on DFS, first, the system workload transition probabilities need to be characterized and derived. Traditional Poisson modeling for the workload can readily provide the transition probabilities for MDP; however, the workload generated in this way varies only lightly [10]. Fluid modeling of exponentially distributed ON/OFF source has been shown to be a good approximation of highly varying workloads [8], [7]. To design a DFS scheme guaranteeing a good power-performance ratio while processing long-range dependent like workloads, we propose to use MDP with fluid analysis of the workload transitions.

The organization of this paper is as follows. First we describe the system model and problem formulation in Section II. Then the two proposed MDP-based frequency controllers, D-FC and C-FC, are derived in Section III, after the exponential ON-OFF fluid analysis of the workload. In Section IV, synthetic traces are used to validate the proposed D-FC and C-FC and real traces are used to benchmark existing DFS.

II. System model and problem formulation

A distributed device/server is modeled by a single-server, singe-queue and single-buffer system for a designated planning horizon. The schematics of the system are shown in Figure 1(a). Requests first arrive at the system buffer stochastically and result in a time-varying workload. A certain computing capacity is required to process the incoming workload to meet the performance target. The computing capacity of the server is a linearly increasing function of the operation frequencies [4], whereas the power consumption is a highly nonlinearly increasing function of the frequencies because of the corresponding voltage scaling [6].

We assume that a frequency controller (FC) exists in the server, which modulates the computation capacity provisioning in a discrete fashion. At every control time point, the FC can set the frequency from a limited and discrete range of values. Here, we define the average workload at the buffer as the target performance metric in the FC. Frequencies are set not only to save power but also to restrict the average buffer workload of that epoch to values below a certain threshold. Once the

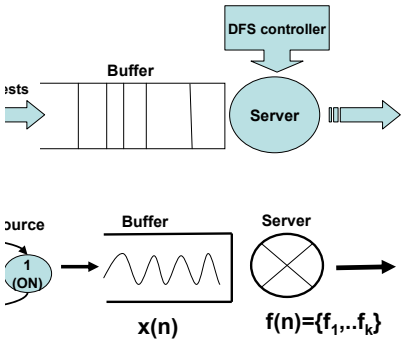


Fig. 1. System schematics (a) and stochastic fluid model (b).

frequency state has been set, it will not be changed until the next decision epoch. Therefore, a prediction mechanism for workload transitions needs to be incorporated into the FC to capture the stochastic nature of the incoming workload. Note that the optimal length of the decision epoch is system- and control-policy-dependent, and that a discussion of its impact would exceed the scope of this paper.

A planned control horizon consists of N finite decision epochs, each of which is indexed by n and takes τ time units. The frequency at n^{th} decision interval is denoted as $f(n)$. We adopt the power consumption per time unit defined in [6] as $P_{\text{fixed}} + P_f \cdot f^3$, where P_{fixed} is the constant term and P_f is a coefficient for a cubic frequency term, provided the server is turned on. Thus, the power consumption of n^{th} control interval is $\{P_{\text{fixed}} + P_f \cdot f(n)^3\} \cdot \tau$. We can therefore summarize the objective of minimizing the total power consumption and the workload constraint in Eq. (1).

$$\begin{aligned} \min \quad & \textstyle\sum_{n=1}^{N} (P_{\text{fixed}} + P_f \cdot f(n)^3) \cdot \tau \\ & E(x; f(n)) \leq \bar{X}, \forall n \\ & f(n) \in \{f_1 \prec f_2 \cdots \prec f_k\}, \forall n \end{aligned} \tag{1}$$

The workload is denoted by x and the average workload of the n^{th} interval, $E(x; f(n))$, is a function $f(n)$ and needs to be less than or equal to a target threshold, $\bar{X}$. The availability of k discrete frequency levels imposes another constraint. Note that with the advance of modern CMOS technology [13], frequency modulation incurs only negligible cost in terms of time and energy.

III. Solution Methodologies

This paper proposes to use MDP techniques to design a state-dependent FC, which prescribes a frequency selection rule based on a defined system state. Note that as the transition of workload depends only on the current state of the system and the frequency selected by the FC in that state, the decision process of the FC is therefore a Markovian decision process. Two steps are required to construct MDP-based FC. The first challenge is to formulate and solve a MDP with the performance constraints of Eq. (1). We define a feasible set of frequencies to accommodate the workload constraint in Subsection III-A, so that the state-of-the-art MDP unconstrained solving algorithms, such as policy iteration [11], can be applied on the resulting unconstrained MDP. The second one is

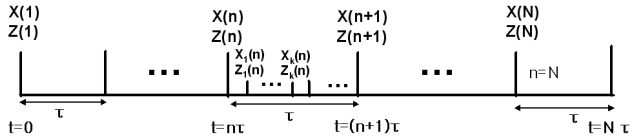


Fig. 2. Control epochs and system state.

to build up the transient analysis for highly varying workload, which will be used to obtain the transition probability in MDP. We adopt a fluid stochastic analysis proposed by Mitra [8] to construct the transient analysis for the long-range dependent like workload considered in Subsection III-B.

A. MDP formulation

We follow the optimization problem in Section II to formulate a finite-horizon MDP problem. It has the objective of energy minimization and defines a feasible frequency set comprising the performance constraints of average workload over N finite control/decision epochs. The process of N epochs is assumed stationary.

As the arrival rate of the workload considered here can be high and bursty, the sequence of incoming requests can be efficiently approximated as a continuous flow of a fluid entering the system. We adopt the stochastic fluid analysis [8], in which the workload fluid enters the buffer according to randomly varying rates governed by a ON-OFF generating source as shown in Fig. 1(b). Thus, we define the system state by (1) the workload fluid level, x , and (2) the ON-OFF state of fluid generator, z, which is a binary variable. Note that in reality only the fluid level (remaining workload) can be observed. The ON-OFF source is essentially a modeling technique to capture the dynamics of the system state, and not observable in reality.

The definitions for state, action, cost function, transition probability, performance constraint, and optimality equations of the proposed MDP are as follows.

D*ecision epoch:* A decision epoch is indexed by $n = \{1, 2 \ldots N\}$, and takes τ units of time. Correspondingly, the decision is executed at time $t = \{1\tau, 2\tau \ldots N\tau\}$ as shown in Figure 2.

S*tate and state space:* Let state $S(n)$ at the beginning of epoch n (at time $t = n\tau$) be defined by $x(n)$ and $z(n)$, where $x(n)$ is the amount of fluid workload in the buffer at the beginning of epoch n and $z(n)$ is the state of the ON-OFF source. We define $x(n)$ in the set of discrete values, $x(n) \in \{0, 1, 2 \ldots F\}$, whereas z is as a binary variable. When $z(n) = 1$, the generating source is in ON state; otherwise. Thus, $S(n) = (x(n), z(n)) \in \{0, 1, 2 \ldots F\} \times \{0, 1\}$.

A*ction and action space:* Let actions $f(n)$ to choose a frequency from k discrete levels $f(n) \in A = \{f_1, f_2, \ldots f_k\}$. Moreover, to construct an unconstrained MDP [11], a feasible set, A', will be used to comply with the performance constraint, $E(x_i, f(n)) = x_i + \sum_j P_{i,j}(f(n)) \cdot x_j \leq \bar{X}$, where $P_{i,j}$ is the transition probability, a function of $f(n)$, defined in the next paragraph. Therefore, $f \in A'$, is a frequency in the feasible action set that obeys the performance constraint: $f \in A' = \{A : x_i + \sum_j P_{i,j}(f(n)) \cdot x_j \leq \bar{X} \ \forall i\}$.

C*ost function:* Let the cost function, $r((x,z),f)$, be defined by the energy consumption.
$r_{f\in A'}(S,f) = r_{f\in A'}((x,z),f) = P_{\text{fixed}} + P_f \cdot f^3$

T*ransition probability:* It defines the probability of changing from state i, $S_i = (x(n) = x_i, z(n) = z_i)$, to state j, $S_j = (x(n+1) = x_j, z(n+1) = z_j)$, during a single control interval of τ units of time, given the action taken at the beginning of the n^{th} interval:
$P_{i,j} = P\{(x(n+1) = x_j, z(n+1) = z_j)|((x(n) = x_i, z(n) = z_i), f)\}$

O*ptimality Equations:* The value function, V_n, at the n^{th} epoch is the minimum of the sum of the current cost ($r((x,z),f)$) and the expected future value (V_{n+1}) function:

$$V_n(x,z) = \min_{f\in A'}\{r((x,z),f) + \sum_{(x'}\sum_{z')\in S} p((x',z')|(x,z),f)v_{n+1}(x',z')\}.$$

Note that this study eventually discretizes the fluid levels for better translatability of the models and solution methodology, even though the fluid is supposed to be continuous. The greater the range of fluid levels considered, the better are the proposed approximation, model, and solution methodologies.

B. Fluid approximations for server workload

The generating source for the fluid workload has an ON and an OFF time following renewal processes and their durations are exponentially distributed. When the ON-OFF source is in the ON state, $z = 1$, it sends the traffic fluid into the buffer at rate γ bytes per unit time, but does not send any traffic during the OFF state ($z = 0$). Here, the buffer is assumed to have infinite waiting capacity and a controllable server capacity, $(f_1, \ldots, f_k)$, i.e, the fluid draining rate, to complete the remaining workload in the buffer as shown in Fig. 1. Note that as we focus on a stationary process, γ is a constant value. In addition to the advantage of being a good approximation to highly varying and long-range dependent traffic, the ON-OFF source can be well modeled by a Continuous Time Markov Chain (CTMC) and a Discrete Time Markov Chain (DTMC), for which existing transient analysis is applicable [3].

1) Continuous Time Markov Chain: Let the source ON and OFF times be exponentially distributed with mean α^{-1} and β^{-1}, respectively. The CTMC generator matrix of the ON-OFF source, M, per time unit is

$$M = \begin{vmatrix} -\beta & \beta \\ \alpha & -\alpha \end{vmatrix}.$$

As the fluid is generated at rate γ during the ON time and 0 during the OFF time, and it is drained out at rate f in between, the buffer is governed by the following drift matrix, D, where I is the identity matrix:

$$D = R - fI = \begin{vmatrix} 0 & 0 \\ 0 & \gamma \end{vmatrix} - fI = \begin{vmatrix} -f & 0 \\ 0 & \gamma - f \end{vmatrix}.$$

Note that f is the control variable at the FC at every control epoch. The higher f is, the more fluid can be drained from the buffer. Essentially, the fluid level, x, changes according to the state of z and the choice of f.

The transition probabilities at time $t = n\tau$, i.e., the beginning of n^{th} epoch, are thus defined as follows:

$H_{i,j}(\tau, x_j, j; x_i, i) = P\{x \le x_j, z(n+1) = z_j | x(n) = x_i, z(n) = z_i\}$.

$$H(\tau, x_j; x_0) = \begin{pmatrix} H_{0,0}(\tau, x_j; x_i) & H_{0,1}(\tau, x_j; x_i) \\ H_{1,0}(\tau, x_j; x_i) & H_{1,1}(\tau, x_j; x_i) \end{pmatrix}.$$

The matrix $H(\tau, x_j; x_i)$ satisfies the following partial differential equation (PDE):

$$\frac{\partial H(\tau, x_j; x_i)}{\partial \tau} + D\frac{\partial H(\tau, x_j; x_i)}{\partial x} = MH(\tau, x_j; x_i), \quad (2)$$

with initial boundary conditions at the beginning of n^{th} epoch, $H(0, x_j; x_i)$.

It is not trivial to solve PDE in Eq. (2). We adopt the solution proposed by Ren and Kobayashi [12], which requires to solve double Laplace transform of H, namely, H^*. In Section IV, we seek a numerical inversion of H^* by means of existing Fourier transform algorithms to obtain H.

As a result, $P_{i,j}$ in MDP can be approximated by the difference of $H\{\tau, x_j; x_i\}$ and $H\{\tau, x_{j'}; x_i\}$, where $x_{j'}$ denotes the adjacent fluid level of x_j:

$$P_{ij} = H_{z_i,z_j}(\tau, x_j; x_i) - H_{z_i,z_j}(\tau, x_{j'}; x_i).$$

2) Discrete Time Markov Chain: To build a DTMC model, first the DTMC transition probability, Q, of the ON-OFF source during a time unit needs to be established. The ON-OFF source at a discrete time point, $k = \{0 \ldots \tau - 1\}$, of the n^{th} epoch, $z_k(n)$, is governed by the transition probability matrix, Q:

$$Q = \begin{pmatrix} Q_{0,0} & Q_{0,1} \\ Q_{1,0} & Q_{1,1} \end{pmatrix}.$$

We applied the uniformization technique [11] on the generating matrix, M, defined in CTMC modeling to obtain Q in DTMC as follows.

$$\text{If } M = \begin{pmatrix} -\beta & \beta \\ \alpha & -\alpha \end{pmatrix} \Rightarrow Q = \begin{pmatrix} 1 - \frac{\beta}{c} & \frac{\beta}{c} \\ \frac{\alpha}{c} & 1 - \frac{\alpha}{c} \end{pmatrix} \quad (3)$$

where $\sup_{s\in(\alpha,\beta)} s \le c \le \infty$. As the Q matrix is just the transition probability per time unit of the ON-OFF source, the transition probability, i.e., the derivation of $P_{i,j}$, in the proposed MDP needs to enumerate all possible combinations of τ ON-OFF changes over the n^{th} epoch. Let $w = \{z_0(n), z_2(n), \ldots z_{\tau-1}(n)\}$ be a sample path of the unit time ON-OFF state at n^{th} control epoch. Therefore, $P(w) = \prod_{k=0}^{k=\tau-1} Q_{(z_k(n), z_{k+1}(n))}$. Let $\Omega = \{w : z_0(n) = z_i, x_0(n) = x_i, z_{\tau-1}(n) = z_j, x_{\tau-1}(n) = x_j\}$ be the set such that a sample path, w, has an initial workload at x_i, with the ON-OFF state as z_i and lets the workload change from x_i to x_j during τ units. Then, $P_{i,j}$ is the product of all $w \in \Omega$, $P_{i,j} = \sum_{w\in\Omega} P(w)$. We then design two FCs based on the optimal decision rules, which are derived from the transition probabilities of the CTMC and DTMC. We call the FCs based on CTMC and DTMC, C-FC and D-FC, respectively.

IV. Experiments

In this section, we present the evaluation of the proposed frequency control schemes, C-FC and D-FC. The pros and cons of both schemes are discussed under synthetic workloads at different frequency modulation windows. Later, we compare C-FC and D-FC with existing DFS algorithms, and show that they achieve a better power-performance ratio.

A. Experimental setup

A frequency control simulator program, written in MATLAB, has been used to generate highly varying workload and modulate the frequency according to the DFS scheme. The workloads used for this evaluation were of two types, namely, synthetic and web traces. Synthetic traces were used to compare C-FC and D-FC in the proposed model, whereas the web traces were used for evaluating the effectiveness of the proposed methodology. The former were generated using an exponential distribution with two parameters, $\alpha = \frac{1}{0.352}$, $\beta = \frac{1}{0.652}$ (the ON and OFF-time parameters, respectively), and $\gamma = 1$ (fluid generating rate). Web traces were obtained from the public domain in the Internet [1]. For convenience of implementation and explanation, we normalize and discretize the remaining workload of the synthetic and web traces into 40 different levels. Eight different levels of operating frequencies, $f_1, \ldots f_8$, corresponding to processing speeds from 0.3 to 0.9, are available for the DFS schemes in FC at every control point. We observed that when running at maximum frequency, the average remaining workload is roughly below the value of 18 for $\gamma = 1$, whereas under operation at the minimum frequency the average remaining workload keeps increasing. Moreover, the remaining workload under maximum frequency is observed to vary greatly that of indicating the potential of the dynamic frequency for energy saving and the challenge involved in achieving an effective power-performance ratio.

B. Proposed C-FC and D-FC on synthetic traces

We first computed the optimal frequencies of C-FC and D-FC corresponding to the aforementioned 40 discrete fluid levels with the workload threshold $\bar{X} = 18$. Figure 3 depicts the optimal frequencies of C-FC and D-FC with $\tau = 5$ and 15 time units, when the fluid generating rate is 1. The top row corresponds to the ON fluid state and the bottom row is to the OFF state. In the implementation of C-FC and D-FC, we obtain the ON and OFF states by estimating differences in fluid levels at decision epochs. Snapshots of simulation result are shown in Fig. 4, where the top row gives the simulation results of the remaining system workload and the bottom row records the frequency modulation at every decision epoch. For a fair comparison, the simulation results are collected from the same random numbers of the ON and the OFF periods. As the optimal frequencies for C-FC and D-FC are the same when $\tau = 5$, the simulation results of C-FC and D-FC are identical, as can be seen in Fig. 4(a). In general, we can observe that the sophisticated workload prediction mechanisms embedded in C-FC and D-FC enable accurate and timing frequency modulation proactively with load changes. A detailed comparison of C-FC and D-FC can be made under different lengths of the control epoch.

1) Length of the control epoch: When the decision epoch is small, $\tau = 5$, only frequency $f1$ and frequency $f8$ are adopted in both C-FC and D-FC. They are separated by certain threshold values of the fluid state, as seen in Fig. 3. Actually, the fluid state values corresponding to the change from $f1$ to $f8$ are the same in C-FC and D-FC. It appears to be the so-called "bang-bang" policy, which operates either at low or at high frequency. Note that D-FC incurs much less computation overhead than C-FC for small τ, whereas C-FC is computationally more efficient for longer τ. On the other hand, during $\tau = 15$, we can observe there are more frequency choices involved in optimal policies, especially for C-FC, because more system states can be obtained in a longer period of time. It turns out that the energy savings due to frequency level modulation are better exploited when higher τ values are considered. Moreover, C-FC is supposed to be more accurate for higher τ as the fluid generator is better modeled by CTMC.

TABLE I
COMPARISONS OF DFSS WITH $\tau = 4$ AND 15.

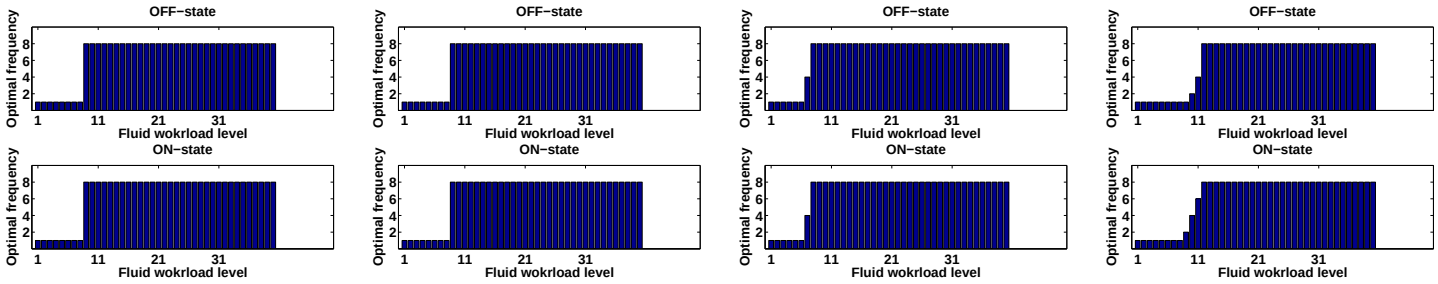
	$\tau = 4$			$\tau = 15$		
Metrics	IBM	Linux	D-FC	IBM	Linux	C-FC
Response time (sec)	0.197	0.117	.050	0.299	0.128	0.085
Energy saving	60%	61%	50%	61%	60%	59%

C. Comparison with existing DFS heuristics

Here we demonstrate the performance comparison of the proposed C-FC and D-FC algorithms with two existing frequency control methods - IBM-ARL, and OnDemand using web traces. The IBM-ARL mechanism [5] monitors the system performance at every decision epoch; it reactively increases the frequency by one level if the system performance is over the target value, and decreases it by one level otherwise. The OnDemand algorithm is also a reactive solution used in Linux [1] OS for frequency setting [9]. Its frequency decrementing policy is the same as that of IBM-ARL, but its frequency incrementing policy is more conservative. If the system performance is observed to be higher than the target value, the frequency will be modulated to the maximum one. Consequently, the OnDemand method is traditionally more conservative towards maintaining performance. To achieve a fair comparison, the system workload is used as an observable system state at FC, and the medium value of system workload is set as the target performance. We calculated the average response time of a request and the normalized energy consumption (over the maximum frequency), which are shown in Table I with $\tau = 4$ and $\tau = 15$ time units. Each time unit equals 1.5 seconds. For the proposed D-FC and C-FC algorithms to work, the history data needs to be fitted into the fluid model to obtain the parameters α, β, and γ. Accordingly, with a suitable level of discrete fluid levels, the optimal policies are computed at the beginning of the DFS planning horizon.

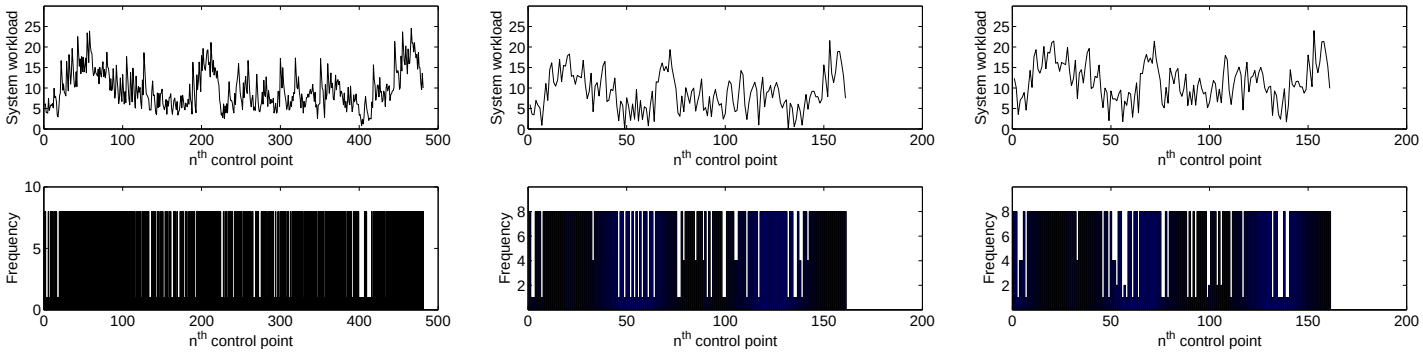
All schemes have similar energy saving ratios, but very different values of the average response time. As IBM-ARL intends to put more emphasis on conserving energy, it has a higher response time than Linux-OnDemand for both control epochs. Moreover, the step-wise frequency increments from IBM-ARL are unable to capture and process sudden and huge workload variations. Consequently, the IBM-ARL scheme is

[1]Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.



(a)D-FC at $\tau = 5$ (b)C-FC at $\tau = 5$ (c)D-FC at $\tau = 15$ (d)C-FC at $\tau = 15$

Fig. 3. Optimal frequency control rules of D-FC and C-FC at $\tau = 5$ and 15, for $\gamma = 1$.



(a) D-FC/C-FC at $\tau = 5$ (b) D-FC at $\tau = 15$ (c) C-FC at $\tau = 15$

Fig. 4. Simulated systems of D-FC and C-FC, at $\tau = 5$ and 15, for $\gamma = 1$.

very sensitive to the control epoch length. On the other hand, the maximum frequency increment in Linux-OnDemand can control the performance of bursty workload better, but may not to be optimal. The proposed C-FC and D-FC have the lowest response time because of their proactive control of the frequency based on the embedded fluid workload analysis in MDP. Well-designed D-FC and C-FC thereby can achieve and maintain a very effective power-performance ratio.

V. Conclusions

We proposed and evaluated two frequency-control schemes, C-FC and D-FC, for highly varying workloads with the objective of minimizing energy consumption and satisfying the constraints of system performance. As the approximation of fluids provides a sound solution for transient analysis needed in the proposed MDP, the proposed DFS schemes can maintain better system performance in defined decision epochs than the existing DFS schemes. The optimal polices computed in our evaluation experiments provide insights into how threshold values should be adopted according to the frequency control window and workload variations. Of them, C-FC is suggested for longer control epochs, whereas D-FC is better suited for finer granularity because of the smaller amount of computation involved. Overall, C-FC and C-FC demonstrate a very effective power-performance ratio on synthetic and on the web traces.

References

[1] Web Caching Project. http://www.ircache.net.
[2] Dimitri P. Bertsekas. *Dynamice Programming*. Prentice, 1988.
[3] G. Bolch, S. Geiner, H. Meer, and K. Trivedi. *Queueing Networks and Markov Chains*. Wiley, 2006.
[4] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing Server Energy and Operational Costs in Hosting Centers. Technical Report CSE-05-002, The Pennsylvania State University, February 2005.
[5] M. Elnozahy, M. Kistler, and R. Rajamony. Energy-Efficient Server Clusters. In *Proceedings of the Second Workshop on Power Aware Computing Systems*, pages 179–196, February 2002.
[6] W. M. Felter, T. W. Keller, M. D. Kistler, C. Lefurgy, K. Rajamani, R. Rajamony, F. L. Rawson, B. A. Smith, and E. Van Hensbergen. On the Performance and Use of Dense Servers. *IBM J. of Res. Develop.*, 47(5):671–688, 2003.
[7] D. Heath, S. Resnick, and G. Samorodnitsky. Heavy Tails and Long Range Dependence in On/Off Processes and Associated Fluid Models. *Math. Oper. Res.*, 23(1):145–165, 1998.
[8] D. Mitra. Stochastic Theory of a Fluid Model of Producers and Consumers Coupled by a Buffer. *Advances in Applied Probability*, 20:646–676, 1988.
[9] V. Pallipadi. Enhanced Intel SpeedStep Technology and Demand-Based Switching on Linux. http://www.intel.com/cd/ids/developer/asmona/eng/195910.htm?page=1.
[10] V. Paxson and S. Floyd. Wide-Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, 3:226–244, 1995.
[11] M. Puterman. *Markov Decision Processes*. Wiley, 1994.
[12] Q. Ren and H. Kobayashi. Transient Solutions for the Buffer Behavior in Statistical Multiplexing. *Performance Evaluation*, 23:65–87, 1995.
[13] K. Roy and S. Prasad. *Low-Power CMOS VLSI Circuit Design*. John Wiley and Sons, New York, 2000.
[14] M. Taqqu, W. Willinger, and R. Sherman. Proof of a Fundamental Result in Self-Similar Traffic Modeling. *Computer Commun. Revi.*, 27:5–23, 1997.