49th IEEE Conference on Decision and Control
December 15-17, 2010
Hilton Atlanta Hotel, Atlanta, GA, USA

Dynamic Power Allocation in Server Farms: a Real Time Optimization Approach

Mohammadreza Aghajani, Luca Parolini and Bruno Sinopoli

Abstract— **This paper concerns the power minimization problem in server farms. The power minimization problem over dynamic power allocation schemes is formally defined and formulated as an optimization problem. It is shown that finding the optimal solution for this optimization problem is not feasible. Inspired by control theory, a well-established method to optimize a cost function over the constraints imposed by the evolution of a dynamical system, called Real-Time Optimization (RTO), is invoked to find a sub-optimal solution for the power minimization problem. The obtained algorithm is simulated and compared with the state-of-the-art optimal static power allocation solution. A considerable improvement in energy consumption is attained for the same quality of service (QoS) level, when dynamic power allocation is used.**

I. INTRODUCTION

Recently, power delivery, electricity consumption, and heat management in server farms have gained considerable attention, because of their quickly rising energy expenditure. In the United States alone, the energy bill due to server farms amounted to about $4.5 billion per year in 2007 [10], while worldwide spending on enterprise power and cooling is reported to be more than $30 billion [19], [17].

This paper focuses on the problem of allocating the available power budget to different servers in order to minimize the total energy consumption while satisfying the required quality of service (QoS). In real world applications, job arrival rates are time-varying. For example, a search engine's load fluctuates considerably during the day, with demand usually higher during office hours than at night. Consequently, use of static power allocation solutions is not satisfactory in many cases.

This paper has two main contributions. First, we formalize the *average power optimization* problem in a server farm and we state it as an optimization problem. Unsurprisingly, the general problem is shown to be hardly tractable both theoretically and computationally. Our second main contribution is a reformulation of the optimization problem as a tractable optimal control one that can be solved using the well-known *two-stage real-time optimization* approach.

The proposed algorithm dynamically changes the power allocated to servers based on the current and predicted incoming load. Due to the non-negligible start-up time of servers, the algorithm has to predict the future processing power demand in order to ensure that sufficient resources are always available. A Markov Modulated Poisson Process (MMPP) is used to describe the time-varying job arrival process, as it qualitatively models the time-varying arrival rate while remaining analytically tractable [9]. The resulting dynamic algorithm is then compared with the state-of-the-art static solutions to show that power consumption can be drastically reduced without compromising the QoS.

The rest of the paper is organized as follows: in Sec. II, the prior work on data center power management as well as the real-time optimization (RTO) method is reviewed. In Sec. III the required assumptions for different system components are stated and the average power minimization problem is formally defined. Section IV discusses our proposed method to obtain a sub-optimal dynamic power allocation for server farms. Section V is devoted to simulations to compare our proposed algorithm with the state-of-the-art static solutions. Finally, conclusions and directions for future work are provided in Sec. VI.

II. PRIOR WORK

Power management techniques span from hardware approaches to decrease processor power consumption to efficient cooling methods. From an implementation viewpoint, the solution operates either locally (chip, server, etc.) [1], [14], [21] or globally (cluster, data center) [17], [10], [18], [19], [7], [16].

Two types of problems can be considered in the power management area [19]. In the *peak power* allocation problems, the QoS is maximized while ensuring that the system does not violate a given peak power budget, where the peak power budget is typically a critical specification of a server farm [8], [5], [10]. In the *average power* allocation problems, the focus is about minimizing the average power required to achieve a certain QoS [18], [1]. Power allocation solutions may be static, or dynamic. In the first case, the power allocated to servers does not vary over the time. In the latter case, the power allocated to servers depends on the number and arrival rate of job requests.

In the process control engineering area, the two-stage real time optimization (RTO) approach is a well-known technique to optimize an "economic" cost in dynamical systems subject to disturbances and uncertainties [3], [2], [6], [15]. The general hierarchical structure of an RTO-based system is shown in Fig. 1. An upper-level real-time optimization layer performs the economic cost optimization at steady state and computes optimal *set-points* for the dynamical system. The validation part ensures that the optimal set-points are achievable and compatible with state or control constraints and passes them to a lower-level dynamic layer. At the dynamic layer, a controller tries to track the set-points and to

Mohammadreza Aghajani, Luca Parolini and Bruno Sinopoli are with the ECE department of Carnegie Mellon University, Pittsburgh, PA {maghajan, lparolin, brunos}@andrew.cmu.edu

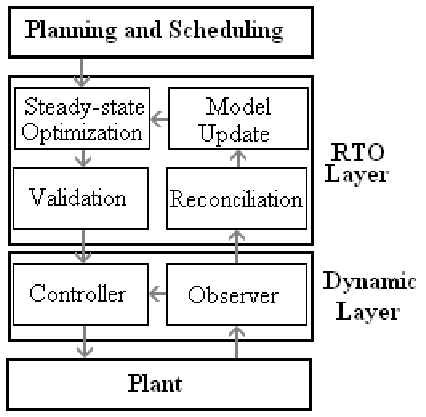

Fig. 1. Hierarchical control structure in real-time optimization systems, [6].

keep the state variables as close to the optimal set-points as possible. Besides, observers at the dynamic layer provide the necessary information to update the steady-state model and re-optimize the set-points. Parameters of both cost function and constraints are provided to the RTOs by the planning and scheduling system.

III. Average Power minimization Problem Definition

Consider a server farm consisting of M similar servers all sharing workload and power supply. Incoming jobs are buffered at the input of an available server according to the task assignment policy. We assume homogeneity in the server farm architecture so that any packet can be routed to any server.

A. Component Model

In order to formally analyze the performance of an arbitrary power allocation scheme, we need models to relate power and frequency in server processors and to describe the job arrival process.

1) Power-vs-Frequency relationship: The frequency at which a processor works may vary based on the power supply voltage and hence the power consumption level. We define such relationship as a function

$$g : \mathscr{P} \to \mathscr{C}; \quad c = g(p), \tag{1}$$

where $\mathscr{P}$ is the set of all possible power consumption levels, $\mathscr{C}$ is the set of frequencies at which a server can operate, p is the server power consumption level, and c is its corresponding frequency. This function depends on processor technology as well as the scaling mechanism used to control this relationship such as DFS (Dynamic Frequency Scaling), DVFS (Dynamic Voltage and Frequency Scaling) and DVFS+DFS [10], [19], [13].

2) Job arrival process: A job arrival model can be derived performing statistical analysis on arrived job traces. There is always a trade off between the accuracy of a stochastic model and its mathematical tractability. Varying-rate processes require additional attention, because models usually lead to mathematically intractable problems.

B. Power Allocation Scheme and Constraints

Let $[0,T]$ be the time interval of interest.

Definition 1- A *static power allocation scheme* P is an ordered set of real numbers $\{p_j \in \mathscr{P}; j = 1,\ldots,M\}$ where p_j is the power allocated to the j^{th} server over the said interval and $\mathscr{P}$ is the set of all possible power consumption levels for a server.

Definition 2- A *dynamic power allocation scheme* F is an ordered set of functions $\{f_j : [0,T] \to \mathscr{P}; j = 1,\ldots,M\}$ where $f_j(t)$ is the power allocated to the j^{th} server at the time t.

A dynamic power allocation scheme can be either open-loop or closed-loop. In open-loop the whole function is determined at time 0, while in closed-loop, the function at time t depends on the observations of the system parameters up to time t. Static power allocation schemes are open loop by definition.

For every power allocation scheme, we can define a cost function E which models the total energy consumed by all servers during the whole time interval. For a static power allocation P we can write

$$E(P) = T \sum_{j=1}^{M} p_j, \tag{2}$$

while for a dynamic power allocation F we have

$$E(F) = \sum_{j=1}^{M} \int_0^T f_j(t)dt. \tag{3}$$

Dynamic power allocations are the main subject of our investigation. In general, we would like to find an optimal power allocation F^* that minimizes (3) while satisfying a set of constraints. We consider two types of constraints: *start-up* constraints and *performance* constraints. The start-up constraints state that a server turned on at a time t cannot start processing immediately. We define *startup* state such a period of time. A set of power allocation schemes that satisfies the startup constraints is denoted by $\mathscr{F}_I$. The performance constraints are usually expressed in terms of performance and throughput parameters, e.g., mean delay time, jitter, total server farm throughput. The set of power allocation schemes that satisfies the performance constraints is denoted by $\mathscr{F}_P$.

C. Problem Formulation

The average power minimization problem over a dynamic power allocation schemes can be defined as

$$\begin{aligned} &\min_F \quad E(F) \\ &F \in \mathscr{F}_I \\ &F \in \mathscr{F}_P. \end{aligned} \tag{4}$$

D. Difficulty in Solving the General Problem

With general models for power-to-frequency relationship and job arrival process, the optimization problem in (4) is extremely hard to solve both theoretically and computationally. First of all, its solution involves a combinatorial search over an extremely large set: for each server s_j, a solution is a function $f_j : [0,T] \to \mathscr{P}$ which is a member of the infinite dimensional set $\mathscr{P}^{[0,T]}$. In practice, we may analyze the system in discrete time with a sampling time τ_s assumed to be very small compared to T to have a reasonably small discretization error. Then, the solutions for each server would be the sequences $f_j(k) \in \mathscr{P}$, $k = 1,\dots,N$, where $N = \frac{T}{\tau_s}$. The solution space, $\mathscr{P}^N$, would still be very large. Furthermore, the number of servers M in a server farm can be very high and therefore, a power allocation solution for the whole server farm would be in the space $\mathscr{P}^{N\times M}$, far from tractable.

The constraints add another level of complication as they are in general non-convex. Consequently, for many practical cases it is not possible to calculate the optimal solution and we need to make some simplifying assumptions and to look for sub-optimal solutions.

IV. An RTO Approach to Average Power Minimization Problem

Let us consider a discrete-time control approach. The resulting power allocation solution will be a piecewise constant function. The incoming job process is also discretized accordingly, which means that all the jobs arrived during a time slot are assumed to have arrived at the beginning of the next time slot.

We also assume that servers only have three states: *on*, *su* (*s*tarting-*u*p), and *off*. In the *off* state a server does not consume power and it does not process jobs ($p = 0, g(0) = 0$). In the *su* state, a server consumes p_{su} watts while it does not process jobs ($g(p_{su}) = 0$). Finally in the *on* state, a server power consumption is p_{on} and its processor works at a nominal frequency c_{on}. The power-vs-frequency relationship can be written as

$$\begin{aligned} &g : \mathscr{P} = \{0, p_{su}, p_{on}\} \to \mathscr{C} = \{0, c_{on}\}; \\ &g = \{(0,0),(p_{su},0),(p_{on},c_{on})\}. \end{aligned} \tag{5}$$

We denote the state of j^{th} server during the k^{th} interval as $S_j(k)$ which takes values in the set {*off, su, on*}. At each time slot k, the number of *on* servers determines the total processing capacity $c(k)$ of the whole server farm

$$c(k) = \sum_{S_j(k)=on} c_{on}. \tag{6}$$

In discrete-time, the arriving job model is given by the number of jobs arrived at each time slot together with their size distribution. In this paper, we assume that all jobs have the same size. Under such an assumption, the incoming job model is described by a random process $\{X(k),\ k \in \mathbb{Z}\}$, where $X(k)$ is the total number of task units arrived at the k^{th} slot. Since all jobs have the same size, the optimal task assignment algorithm is the one which assigns each task unit to the least busy server.

A Markov Modulated Poisson Process (MMPP) is used to model the varying-rate incoming job process $\{X(k)\}$ [9]. The discrete-time MMPP consists of a finite-state discrete-time Markov chain whose states are chosen from the set $\mathbb{S} = \{s_1, s_2, \dots, s_n\}$ with a transition matrix P and a set of arrival rates $\{\lambda_1, \lambda_1, \dots, \lambda_n\}$. The state of the Markov chain during the k^{th} slot, denoted with $Z(k)$, determines the rate at which jobs arrive at the server farm, and we call it *Markov chain rate*. Let s_i be the state of the Markov chain in the k^{th} slot, i.e. $Z(k) = s_i$. Then, the distribution of jobs arriving at the server farm during the same slot has a Poisson distribution with rate λ_i. Job arrival rates change randomly according to the transition matrix P. We set the discretization sampling time to be the amount of time it takes for an *on* server to process a task unit. The frequency c_{on} is then normalized to 1 job units per time slot.

When a server is switched on, it cannot immediately start processing the assigned tasks. Let T_s be the number of intervals that a server requires in order to transition from the *off* state to the *on* state. We assume that the transition from the *on* state to the *off* state takes a negligible amount of time.

Given the above constraint, the state of servers at the beginning of the k^{th} slot is determined by the sequence of the switching commands, i.e. the commands to switch the servers on and off, during all time slots before k. The problem (4) can now be reformulated as

$$\begin{aligned} &\min_u \quad E(F(u)) \\ &F(u) \in \mathscr{F}_P, \end{aligned} \tag{7}$$

where $F(u)$ is the power allocation scheme when the switching command u is applied.

Since the decision to turn on a server yields its effect only T_s interval later, the future status of the system need to be taken into account to make a decision at the current time.

The performance constraint we consider is the mean delay. Mean delay is the expected time it takes to complete a job from its arrival at the server farm. A power allocation scheme is acceptable only if the mean delay time is less than a certain predetermined value T_D.

A. Optimal Control formulation

Let $l(k)$ be the number of jobs in the server farm during the k^{th} slot. The number of jobs arriving at the server farm during the k^{th} interval is the uncontrollable, stochastic input, while the switching command is the controllable input. We can describe the evolution $l(k)$ as

$$l(k+1) = l(k) + X(k) - min\Big\{l(k),\, c^u(k)\Big\}, \tag{8}$$

where $c^u(k)$ is the service rate of the server farm at the time k when the switching command control u is applied. The cost function (3) can be written as

$$E(u) = \sum_{k=1}^{N} \left(\sum_{S_j(k)\in ON(u,k)} p_{on} + \sum_{S_j(k)\in SU(u,k)} p_{su} \right), \tag{9}$$

where for each time slot k, $ON(k,u)$ is the set of *on* servers and $SU(k,u)$ is the set of *starting-up* servers where the switching control commands u is applied.

Performance constraints in (7) can be written in terms of the state variables l_j and the control input u

$$D(u) = \mathbb{E}\left[\frac{\sum_k 1 + l(k)}{\sum_k X(k)}\right], \tag{10}$$

where the expectation is taken over the incoming job distribution. Using (9) and (10), the optimal power allocation problem in (4) can be rewritten as

$$\begin{aligned} \min_u \quad & E(u) \\ D(u) \leq T_D. & \end{aligned} \tag{11}$$

B. Real-Time Optimization approach

In our scenario, the job arrival rate is time varying and unknown a priori. The model update part uses statistical properties of the incoming job process as well as the observed data to provide an estimated value of the job arrival rate. In the steady-state optimization part, the cost function is optimized based on the steady-state behavior of a queueing theoretic model of the server farm, given the estimated arrival rate. The optimal set-points are passed to the dynamic controller in terms of the required processing capacity as well as the corresponding steady-state queue length, and the switching command decisions are made based on these set-points.

For a given job arrival rate λ and for any number of active servers c, the dynamic equation (8) can be viewed as a discrete-time G/D/c queueing system [12]. In this queueing model, the number of incoming jobs in one slot has a Poisson distribution of parameter λ, the service time for each job is deterministic, i.e. 1, and c is the number of servers.

The steady-state behavior of a discrete-time G/D/c queueing systems is well studied [12], [11], [20]. The probability generating function $\mathscr{L}(z)$ (PGF) of l is

$$\mathscr{L}(z) = c(1-\rho)\frac{(z-1)\mathscr{X}(z)}{z^c - \mathscr{X}(z)}\prod_{i=1}^{c-1}\frac{z-z_i}{1-z_i}, \tag{12}$$

where $\rho = \frac{\lambda}{c}$, $\mathscr{X}(z)$ is PGF of the input distribution X, and z_i, $1 \leq i \leq c-1$ are $c-1$ zeros of $z^c - \mathscr{X}(z)$ inside the unit disk $\{z : |z| < 1\}$ (proved to exist by Riuchè's theorem [4]). The mean value $\bar{l}$ can is given by

$$\bar{l} = \mathscr{L}'(1) = \lambda - \frac{(c-1)-\rho^2 c}{2(1-\rho)} + \sum_{i=1}^{c-1}\frac{1}{1-z_i}. \tag{13}$$

Accordingly, the mean delay time D can be computed using *Little's* theorem [20] yielding $D = \frac{\bar{l}}{\lambda}$. For a fixed input rate λ, the optimal (most economic) valid service rate is the smallest value of $\bar{c}$ for which the stability condition holds ($\bar{c} > \lambda$) and the corresponding delay time D satisfies the performance condition $D \leq T_D$.

C. Dynamic Control approach

Given the optimal set-point $(\bar{c}, \bar{l})$, the dynamic control problem is a tracking problem which can be addressed using classical control tools. We use the following controller

$$u = \left\lfloor \left(\alpha(\hat{c} - \bar{c}) + \beta(\hat{l} - \bar{l})\right) \right\rfloor, \tag{14}$$

where u is the resulting switching command (the number of servers which should be turned on or off, according to the sign of u), $\hat{c}$ and $\hat{l}$ are predicted service rate and system content, respectively, at T_s slots ahead if no switching is applied, α and β are coefficients that can be appropriately tuned, and $\lfloor\cdot\rfloor$ is the floor operator.

D. Observation and Model Update

In an MMPP the arrival rate changes randomly according a transition matrix P. The state of the Markov chain rate at the current time is unknown and a probability distribution over all possible states describes our knowledge about the current arrival rate.

Let the vector $\boldsymbol{\mu}(k+\tau|k) \in \mathbb{R}^n$ be the probability distribution over all n possible states of Markov chain rate at the time slot $k+\tau$, conditioned on all the observations up to time k. The initial distribution $\boldsymbol{\mu}(0|0)$ is assumed to be known, and the current time distribution $\boldsymbol{\mu}(k|k)$ can be derived recursively from the distributions in the past, using the following algorithm:

Step 1 - Time Update: Having $\mu(k|k)$ in hand, by the Chapman-Kolmogorov equation for discrete-time Markov chains we can write

$$\boldsymbol{\mu}(k+1|k) = \boldsymbol{\mu}(k|k)P. \tag{15}$$

We call (15) the *time update* equation.

Step 2 - Measurement update: Once the observation at the time $k+1$ is available, we can use the number of arrived jobs to correct the probability distribution on Markov chain rate states. The corrected distribution $\boldsymbol{\mu}(k+1|k+1)$ is the a posteriori probability distribution given the number of arrived jobs $X(k+1)$ at the time $k+1$. Assume the observer reports that x jobs have arrived during the time slot $k+1$, i.e. the event $\{X(k+1) = x\}$ is observed, we can write

$$\begin{aligned} & P[Z(k+1) = s_j \mid X(k+1) = x] && (16)\\ & \quad = \frac{P[X(k+1) = x|Z(k+1) = s_j]P[Z(k+1) = s_j]}{P[X(k+1) = x]} \\ & \quad = \frac{e^{-\lambda_j}\frac{\lambda_j^x}{x!}\mu_j(k+1|k)}{\sum_{i=1}^n e^{-\lambda_i}\frac{\lambda_i^x}{x!}\mu_i(k+1|k)}, && (17) \end{aligned}$$

where $\mu_j(k+1|k)$ is the j^{th} element of the vector $\boldsymbol{\mu}(k+1|k)$.

The arrival rate used in the steady-state optimization part of the control approach is $\hat{\lambda}(k+T_s|k)$, i.e. the expected arrival

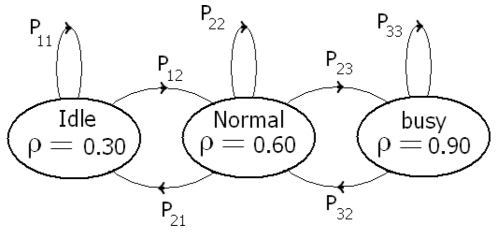

Fig. 2. Markov chain rate in incoming job MMPP model

rate T_s time steps ahead. This prediction can be computed as

$$\begin{aligned}\hat{\lambda}(k+T_s|k) &= \mathbb{E}[X(k+T_s)] \qquad (18)\\ &= \sum_{j=1}^{n} \mathbb{E}[X(k+T_s)|Z(k+T_s)=j]\,\mathbb{P}[Z(k+T_s)=j]\\ &= \sum_{j=1}^{n} \lambda_j\,\mu_j(k+T_s|k),\end{aligned}$$

where the probability distribution $\boldsymbol{\mu}(k+T_s|k)$ can be computed as

$$\boldsymbol{\mu}(k+T_s|k) = \boldsymbol{\mu}(k|k)P^{T_s}. \qquad (19)$$

V. Simulation Results

In this section a server farm with $M = 2000$ server is considered. The traffic is generated via a Montecarlo approach. Since computation of the optimal solution for the dynamic power allocation problem (7) is infeasible, our dynamic solution is compared to the state-of-the-art optimal static solution.

The performance of our proposed solution depends on a number of parameters, especially on the starting up period T_s, start up energy consumption level P_{su} and the Markov chain rate transition matrix P. We examine the effect of each of these parameters and describe a condition under which the proposed dynamic strategy provides significant energy savings.

Figure 2 shows the Markov chain rate and the corresponding job arrival rates in the MMPP model we used in our simulation. The Markov chain rate has three states corresponding to idle, normal and busy periods of a server farm, with the incoming workload rates equal to 30%, 60% and 90% of maximum processing power of the server farm, respectively.

The change in the load arrival rate can be measured by the mean run time τ of the Markov chain rate, i.e. the expected time that the Markov chain rate spends in the same state before jumping to another. The number of successive time steps that the Markov chain rate stays in the j^{th} state has a geometric distribution with parameter $1-P_{jj}$. So the mean run time τ_j for the j^{th} state is $\frac{1}{1-P_{jj}}$. Averaging over all states, the mean run time τ is

$$\tau = \sum_{j=1}^{M} \mu_j \tau_j = \sum_{j=1}^{M} \frac{\mu_j}{1-p_{jj}}, \qquad (20)$$

where μ_j is the j^{th} element of the steady-state distribution $\boldsymbol{\mu}$ of the Markov chain rate.

The mean delay time obtained by the RTO solution is depicted in Fig. 3 for different values of the ratio of $\frac{T_s}{\tau}$. The allowable delay time is set to be 3 time slots and the dynamic power allocation algorithm tries to keep the mean delay below this threshold, while minimizing the total power consumption. As depicted in Fig. 3, the mean delay is successfully kept below the threshold approximately for $\frac{T_s}{\tau} < 0.5$, but the algorithm fails for larger ratio values as predicted above. Our algorithm is then compared with that of the state-of-the-art optimal static power allocation scheme. For each value of $\frac{T_s}{\tau}$, our proposed dynamic algorithm and the optimal static algorithm are compared and the total energy needed to attain a same performance level in each case is compared. The energy consumption improvement provided by the dynamic algorithm is shown in Fig. 4. The proposed dynamic power allocation algorithm offers the same mean delay time with 30% less consumed energy, when the incoming rate changes are sufficiently slower than the servers' dynamics ($\frac{T_s}{\tau} \ll 1$), Fig. 4. This energy consumption improvement decreases as the workload rate changes becomes faster and finally for very abrupt changes in arrival rate (i.e. large values of $\frac{T_s}{\tau}$), the static power allocation strategy might be preferable.

The behavior of the system can be justified considering the energy saved when turning off idle servers and the energy spent by starting up servers which cannot immediately process jobs. When the changes on the arrival job rate are slow compared to the server start-up time, the saved energy is dominant so the dynamic power allocation algorithm provides better performance than the static one. On the other hand, when the changes on the arrival job rate are on the same scale of the server start-up time, the energy spent on starting up servers is dominant and this causes the performance of the dynamic algorithm to degrade considerably compared to its static counterpart.

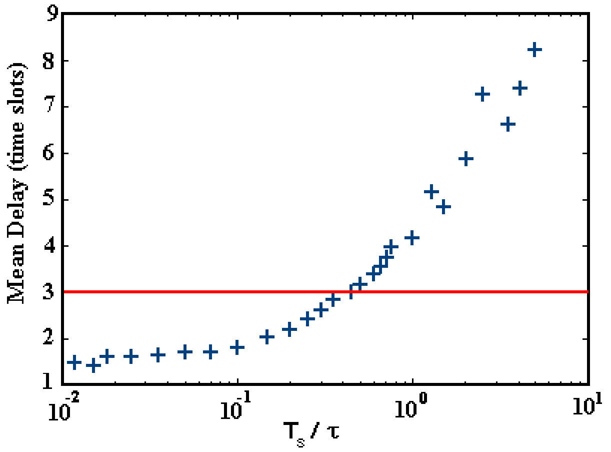

Fig. 3. Mean delay time under dynamic power allocation for different values of $\frac{T_s}{\tau}$ ratio. The allowable delay time is set to be 3 time slots.

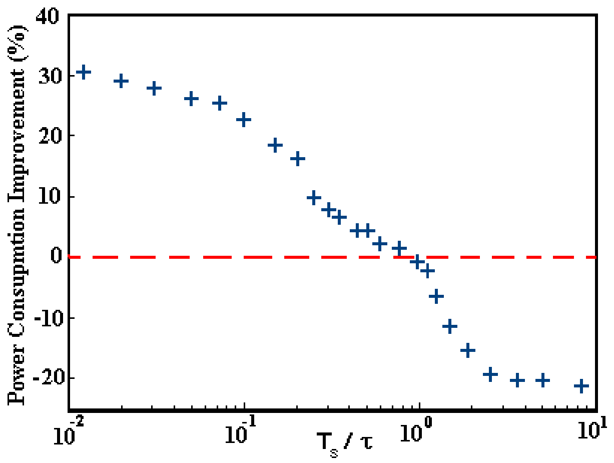

Fig. 4. Improvement in total energy consumption made by proposed dynamic algorithm comparing to optimal static algorithm for different values of $\frac{T_s}{\tau}$ ratio. when the same QoS level is attained.

Slowly varying job arrival rate is usually the dominant case for the server farm applications. The starting up time T_s is usually in the order of seconds to minutes, while the time scale of incoming job rate is in the hours time-scale (a server farms may have busy hours and idle hours in a day). So for many practical applications, we expect a 10%-20% saving in consumed energy.

VI. Conclusion and Future Work

In this paper the average power minimization over dynamic power allocation schemes is formally defined and formulated in terms of an optimization problem. Considering the infeasibility of the general problem, a sub-optimal method is proposed. Inspired by optimal control theory, the well-known two-stage real-time optimization method is invoked and the original problem is split into two separate problems: a steady-state optimization problem which provides the optimal set-points, and a tracking problem to reach the provided optimal set-points.

The performance of the proposed control algorithm is shown via a computer simulation and the effect of different system parameters is investigated. Under a condition called *slowly varying* job arrival rate, typically satisfied in server farms, the proposed algorithm guarantees the allowable mean delay and considerably improves upon the state-of-the-art optimal static power allocation with the same overall energy consumption.

There are a number of simplifying assumptions made in this study which can be generalized in future work. Without fundamental changes, the proposed control approach can be extended to the cases in which the servers have more than two power consumption levels and corresponding processing frequencies, as long as there are finite number of such levels. Removing the assumption about splitting the incoming jobs into equally sized task units require the development of a task assignment policy. In such a case, the development of a task assignment policy and of a power allocation scheme should be jointly considered under the same optimization problem.

References

[1] T. F. Abdelzaher, K. G. Shin, and N. Bhatti. Performance guarantees for web server end-systems: A control-theoretical approach. *IEEE Trans. Parallel Distrib. Syst.*, 13(1):80–96, 2002.

[2] E. M. B. Aske, S. Strand, and S. Skogestad. Coordinator MPC for maximizing plant throughput. *Computers & Chemical Engineering*, 32(1-2):195 – 204, 2008.

[3] L. T. Biegler and J. B. Rawlings. Optimization approaches to nonlinear model predictive control. *Chemical Process Control*, 1991.

[4] H. Bruneel and I. Wuyts. Analysis of discrete-time multiserver queueing models with constant service times. *Operations Research Letters*, 15(5):231 – 236, 1994.

[5] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle. Managing energy and server resources in hosting centers. *SIGOPS Oper. Syst. Rev.*, 35(5):103–116, 2001.

[6] S. Engell. Feedback control for optimal process operation. *Journal of Process Control*, 17(3):203 – 219, 2007.

[7] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In *34th Annual International Symposium on Computer Architecture*, 2007.

[8] M. E. Femal and V. W. Freeh. Boosting data center performance through non-uniform power allocation. *Autonomic Computing, International Conference on*, 0:250–261, 2005.

[9] W. Fischer and K. Meier-Hellstern. The Markov-modulated poisson process (MMPP) cookbook. *Performance Evaluation*, 18(2):149 – 171, 1993.

[10] A. Gandhi, M. Harchol-Balterr, R. Das, and C. Lefurgy. Optimal power allocation in server farms. In 11th *International joint conference on Measurement and modeling of computer systems*, 2009.

[11] P. Gao, S. Wittevrongel, and H. Bruneelg. Delay and partial system contents for a discrete-time g-d-c queue. *4OR: A Quarterly Journal of Operations Research*, 6:279–290, 2008.

[12] P. Gao, S. Wittevrongel, J. Walraevens, M. Moeneclaey, and H. Bruneel. Calculation of delay characteristics for multiserver queues with constant service times. *European Journal of Operational Research*, 199(1):170 – 175, 2009.

[13] A. Leizarowitz. Infinite horizon autonomous systems with unbounded cost. *Applied Mathematics & Optimization*, 13:19–43, 1985.

[14] J. Mao, C. G. Cassandras, and Q. Zhao. Optimal dynamic voltage scaling in energy-limited nonpreemptive systems with real-time constraints. *IEEE Transactions on Mobile Computing*, 6:678–688, 2007.

[15] T. Marlin and A. N. Hrymak. Real-time operations optimization of continuous processes. *AICHE SYMPOSIUM SERIES*, 316:156–164, 1997.

[16] L. Parolini, N. Tolia, B. Sinopoli, and B. H. Krogh. A cyber-physical systems approach to energy management in data centers. In *International conference on cyber-physical systems*, Apr. 2010.

[17] C. Patel and P. Ranganathan. Enterprise power and cooling, Oct. 2006. ASPLOS Tutorial.

[18] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath. Load balancing and unbalancing for power and performance in cluster-based systems. In *Workshop on Compilers and Operating Systems for Low Power (COLP)*, 2001.

[19] R. Raghavendra, P. Ranganathan, V. Talwar, Z.Wang, and X. Zhu. No power struggles: Coordinated multi-level power management for the data center. In *13th International Conference on Architectural Support for Programming Languages and Operating Systems*, 2008.

[20] T. G. Robertazzi. *Computer Networks and Systems: Queueing Theory and Performance Evaluation*. Springer-Verlag New York, Inc., 2000.

[21] A. Varma, B. Ganesh, M. Sen, S. R. Choudhury, L. Srinivasan, and B. Jacob. A control-theoretic approach to dynamic voltage scheduling. In *International conference on compilers, architecture and synthesis for embedded systems*, 2003.