A PERFORMANCE COMPARISON OF DYNAMIC VS. STATIC LOAD BALANCING POLICIES IN A MAINFRAME-PERSONAL COMPUTER NETWORK MODEL

Hisao Kameda, Said Fathy El-Zoghdy, and Jie Li

September 14, 2001

ISE-TR-01-183

H. Kameda is with the Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba Science City, Ibaraki 305-8573, Japan. Tel: +81–298–53–5539 Fax: +81–298–53–5206 Email: kameda@is.tsukuba.ac.jp

Said F. El-Zoghdy is with the Doctoral Program in Engineering, University of Tsukuba, Tsukuba Science City, Ibaraki 305-8573, Japan. Tel: +81–298–53–5156 Fax: +81–298–53–5156 Email: said@osdp.is.tsukuba.ac.jp

Jie Li is with the Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba Science City, Ibaraki 305-8573, Japan. Email: lijie@is.tsukuba.ac.jp

A Performance Comparison of Dynamic vs. Static Load Balancing Policies in a Mainframe – Personal Computer Network Model

Hisao Kameda,* Said Fathy El-Zoghdy,† and Jie Li‡

Abstract

Distributed and networked computers can share job processing in the event of overloads. Load balancing involves the distribution of jobs throughout a system of networked computers, thus increasing processing capacity of the system without having to obtain additional or faster computer hardware. Load balancing policies may be either static or dynamic. In general, dynamic policies are more complex and have more overhead than static ones, and truly optimal dynamic policies are known only for special systems. This study focuses on performance comparison between static and dynamic load balancing policies in a distributed computer system where truly optimal solutions of both dynamic and static policies have been characterized. The system consists of two types of service facilities, a Mainframe node and an unlimited number of Personal Computer nodes. Overheads due to the policies are assumed to be negligible. We investigate the $[L, q]$ threshold rule that has been already proposed as a dynamic load balancing policy. The results show that, in the model examined, the dynamic load balancing policy outperforms the static one in the system mean response time, at most about 30 percent. In addition, we see that, the minimum system mean response time is obtained by the dynamic load balancing policy, i.e., the $[L, q]$ threshold) policy with $q = 0$ and the suitable selection of the other threshold parameter L.

keywords Communication Networks, Distributed/Parallel Computer Systems, Computer Systems Performance, Queueing Theory, Optimization, Load Balancing Policies.

1 Introduction

Distributed computing systems, such as networks of workstations or mirrored sites on the World Wide Web, face the problem of using their resources effectively. If some hosts

*H. Kameda is with the Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba Science City, Ibaraki 305-8573, Japan. Tel: +81–298–53–5539 Fax: +81–298–53–5206 Email: kameda@is.tsukuba.ac.jp

†Said F. El-Zoghdy is with the Doctoral Program in Engineering, University of Tsukuba, Tsukuba Science City, Ibaraki 305-8573, Japan. Tel: +81–298–53–5156 Fax: +81–298–53–5156 Email: said@osdp.is.tsukuba.ac.jp

‡Jie Li is with the Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba Science City, Ibaraki 305-8573, Japan. Email: lijie@is.tsukuba.ac.jp

lie idle while others are extremely busy, system performance may fall significantly. To prevent this, load balancing is often used to distribute the workload [21]. A large number of load balancing policies have been proposed to improve the performance of distributed/parallel systems (e.g., to minimize the mean response time of a job, to maximize the processing capacity of the system) by efficiently utilizing the processing power of the entire system. This is done by redistributing the workload among nodes. Although a communication delay is incurred in transferring a job from one node to another, the performance of a distributed computer system can generally be improved by an effective load balancing policy [2, 12, 13, 14, 15]. Load balancing policies may be either static or dynamic.

Static load balancing policies [3, 4, 6, 9, 10] use only the statistical information on the system (e.g., the average behavior of the system) in making load-balancing decisions, and their principal advantage is lower overhead cost needed to execute them and their simplicity in implementation and their mathematical tractability. They do not, however, adapt to fluctuations in workload. Under a situation where the system workload is statistically balanced, some computers may be heavily loaded at a given instant (hence suffering from performance degradation), while others are idle or lightly loaded.

On the other hand, dynamic load balancing policies [2, 3, 10, 16, 17, 18, 19, 20] attempt to dynamically balance the workload reflecting the current system state and are therefore thought to be able to further improve the system performance. Thus, it would be thought that, compared to static ones, dynamic load balancing policies are better able to respond to system changes and to avoid those states that result in poor performance. Obviously, the disadvantages of dynamic load balancing policies is that these policies are more complex than their static counterparts, in the sense that they require information on the runtime load and activities of state collection. The effect of occasionally poor load balancing decisions and the potential for instability in dynamic load balancing because of the inherent inaccuracy of system state information have been studied in [21].

In this paper, we consider *dynamic and static overall optimal policies* whereby job scheduling is determined so as to minimize the system mean response time. The goal of this paper is to examine to what extent the optimal dynamic load balancing policy outperforms the static one by an exhaustive numerical investigation on a model for which both policies are analytically studied. Optimal static load balancing policies have been analytically studied in a variety of models for distributed computer systems [4, 5, 6, 7, 9]. On the other hand, as far as we know, optimal dynamic load balancing policies have been studied only in very specific models: one is that of using an M/M/*m* queueing model [2], and another is what is analytically studied in [1]. The latter is the model studied here that consists of a Mainframe node Q_{MF} and an unlimited number of Personal Computer nodes Q_{PC}. The dynamic load balancing policy considered in the model is the $[L, q]$ threshold rule whereby a job arriving at the Q_{PC} node is forwarded to the Q_{MF} with probability 1 if the number of jobs staying at the Q_{MF} node is less than L, with probability q if the number equals L, and otherwise is processed by the Q_{PC} node. The model allows us to have exhaustive numerical investigation to gain insight into the problem. The objective of both policies is to minimize the overall system mean response time. We do not take account of the difference in the overheads due to the policies.

While there have been some studies of performance comparison between dynamic and static load balancing policies in more sophisticated models where overheads are considered, the truly optimal dynamic policy is not accurately obtained in contrast to the model considered here [3, 9]. The results obtained here show that, in the model examined, the dynamic load balancing policy outperforms the static one in the system mean response time, at most about 30 % and for the range of parameter values such that the arrival rate is close to the processing rate of the Mainframe node. Another remarkable finding is that the minimum system response time is achieved by the $[L, q]$ threshold rule with $q = 0$. That is, we need to choose only the proper value of L with q fixed to be 0 in finding the set of parameter values of the threshold rule that gives the minimum mean system response time.

This paper is organized as follows. Section 2 describes the system model of this paper. Section 3 presents two optimal load balancing policies: static and dynamic. Section 3.2 shows that the minimum mean system response time is achieved by the dynamic load balancing policy, i.e., the $[L, q]$ threshold policy with $q = 0$, and presents the algorithm used to obtain the optimal threshold parameter L. Section 4 describes the results of numerical examination. Finally, Section 5 summarizes this paper.

2 The System Model

We consider the model of a distributed computer system that consists of two types of service facilities, a Mainframe node Q_{MF} and unlimited number of Personal Computer nodes Q_{PC}, both of which are connected by a communication network. We call this system model an *MF-PC network model.* We assume that the expected communication delay between the Q_{MF} node and the Q_{PC} node is negligible. Jobs arrive at the system according to a time-invariant Poisson process, i.e. inter-arrival times of jobs are independent, identically and exponentially distributed with mean $1/\lambda$. Simultaneous arrivals are excluded. A job arriving at the system may be processed either by the Q_{MF} node or by the Q_{PC} node according to load balancing policies. We assume that the service rate at Q_{MF} is μ and that its service discipline is first-come-first-served (FCFS), or processor sharing whereby the service rate for each job equals $v(n) = \mu/n$ when the number of jobs in the Q_{MF} node is n. The Q_{PC} node offers a fixed expected service time θ^{-1}. In the Q_{PC}, service starts immediately upon admission, and thus the mean response time is identical to the service time. We assume that at both Q_{MF} and Q_{PC}, service times are independent, identically and exponentially distributed.

3 Two Optimal Load Balancing Policies

In the following two subsections, we present optimal static and dynamic load balancing policies and their solutions.

3.1 Optimal Static Load Balancing Policy

In this policy, the decision of transferring a job from one node to another does not depend on the state of the system, and hence is *static* in nature. Also, we assume that a job transferred from one node to another receives its service there, and is not further transferred.

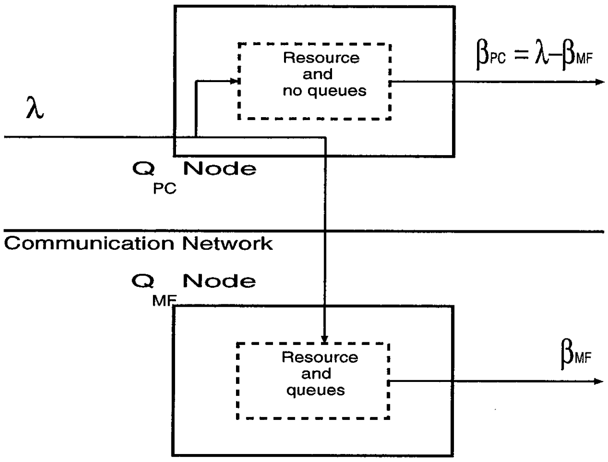

Figure 1: A model of an MF-PC network system

In this section, we consider an optimal static load balancing policy that determines the optimal load at each node so as to minimize the mean job response time in our system model.

We use the following notation:

- β_{MF}: Job processing rate (load) at the Q_{MF} node.
- $F_{MF}(\beta_{MF})$: Expected delay of a job processed at the Q_{MF} node.

$$F_{MF}(\beta_{MF}) = \begin{cases} \dfrac{1}{\mu - \beta_{MF}} & if\ \beta_{MF} < \mu, \\ \infty & \text{otherwise.} \end{cases}$$

The problem of minimizing the mean system job response time is expressed as

$$\begin{aligned} & \text{minimize } D(\beta_{MF}) \\ = \ & \frac{1}{\lambda}[\beta_{MF}F_{MF}(\beta_{MF}) + (\lambda - \beta_{MF})\theta^{-1}] \end{aligned} \tag{1}$$

with respect to β_{MF} such that $0 \le \beta_{MF} \le \lambda$.

Define β_0 ($0 \le \beta_0 < \mu$) such that

$$\frac{\mu}{(\mu - \beta_0)^2} = \theta^{-1}.$$

The optimal β_{MF} is given as follows:

$$\beta_{MF} = \begin{cases} \beta_0 & if\ \beta_0 < \lambda, \\ \lambda & if\ \lambda \le \beta_0. \end{cases}$$

3.2 Optimal Dynamic Load Balancing Policy

By this policy, each arriving job may observe the current load in the Q_{MF} node, and then choose whether to join the shared mainframe or to remain at the Q_{PC} node. Also, the goal is to minimize the mean system response time per job. Observing that the mean system response time does not depend on the service discipline in the Q_{MF} node (PS, FCFS, etc.), the problem reduces to that of a standard queueing control.

A class of threshold load balancing policies have been shown to be useful when jobs are completely independent and consists of single threads of control. This situation is fairly common in networks of workstations. Such threshold policies contain control parameters (e.g. threshold values and transfer probability for every host), that require fine-tuning in order to yield optimal or near optimal performance. For the work on threshold policies, the reader is referred to [2, 11, 12].

We use the $[L, q]$ threshold rule as the dynamic load balancing policy. In this rule, an arriving job will go to the Q_{MF} node with probability of, respectively, 0, q, and 1, if the job finds that the Q_{MF} node has, more than, equal to, and less than, L jobs. We consider a formula $E\left[W_{[L,q]}\right]$ for the mean response time of the system with respect to $[L, q]$ threshold rule and minimize $E\left[W_{[L,q]}\right]$. The mean response time of a job arriving at the system with threshold $[L, q]$, $E\left[W_{[L,q]}\right]$, is obtained as follows:

$$E\left[W_{[L,q]}\right] = P\theta^{-1} + Q\lambda^{-1},$$

where, if $\rho \neq 1$ (i.e. $\lambda \neq \mu$),

$$\begin{aligned} P &= P_0(1 - q + q\rho)\rho^L, \\ Q &= P_0\rho\frac{(-(L+1)\rho^L)(1-\rho) + (1-\rho^{L+1})}{(1-\rho)^2} \\ &\quad +(L+1)P_0 q\rho^{L+1}, \\ P_0 &= \frac{1-\rho}{1-\rho^{L+1}(1-q) - q\rho^{L+2}}, \end{aligned}$$

and if $\rho = 1$ (i.e. $\lambda = \mu$),

$$P = \frac{1}{L+1+q}, \quad Q = \frac{(L+1)(L+2q)}{2(L+1+q)}.$$

(For the derivation of the above, see Appendix A.)

Proposition 1 *The mean system response time is minimized by the threshold policy with the value of threshold parameter $q = 0$.*

proof: Note that the $[L, 1]$ threshold policy is identical with the $[L+1, 0]$ threshold policy. It is sufficient to show that, given λ, μ, θ and L, $E\left[W_{[L,q]}\right]$ is monotonically non-decreasing or non-increasing in $q \in [0, 1]$. That is, either $\frac{\partial}{\partial q}E\left[W_{[L,q]}\right] \geq 0$ for all $q \in [0, 1]$, $\frac{\partial}{\partial q}E\left[W_{[L,q]}\right] \leq 0$ for all $q \in [0, 1]$, or $\frac{\partial}{\partial q}E\left[W_{[L,q]}\right] = 0$ for all $q \in [0, 1]$.

It can be shown as follows. Given λ, μ, θ and L, we have the following two distinct cases:

- Case (1): $\rho = 1$ (i.e. $\lambda = \mu$) and $q \in [0, 1)$
- Case (2): $\rho \neq 1$ (i.e. $\lambda \neq \mu$) and $q \in [0, 1)$

Case (1): $\rho = 1$ (i.e. $\lambda = \mu$) and $q \in [0, 1]$

$$E\left[W_{[L,q]}\right] = \frac{1}{(L+1+q)}\theta^{-1} + \frac{(L+1)(L+2q)}{2(L+1+q)}\lambda^{-1},$$

$$\frac{\partial E}{\partial q} = \frac{-2\lambda + (2+3L+L^2)\theta}{2\lambda\theta(L+1+q)^2}. \tag{2}$$

Case (2) $\rho \neq 1$ (i.e. $\lambda \neq \mu$) and $q \in [0, 1]$

$$E\left[W_{[L,q]}\right] = P\theta^{-1} + Q\lambda^{-1},$$

where

$$\begin{aligned}
P_0 &= \frac{1-\rho}{1-\rho^{L+1}(1-q)-q\rho^{L+2}}, \\
Q &= P_0\rho\frac{(-(L+1)\rho^L)(1-\rho)+(1-\rho^{L+1})}{(1-\rho)^2} \\
&\quad +(L+1)P_0q\rho^{L+1}, \\
P &= P_0(1-q+q\rho)\rho^L.
\end{aligned}$$

Hence,

$$\frac{\partial E}{\partial q} = \frac{\rho^L(C_1-C_2)}{\lambda\theta(1+\rho^{L+1}(q-1)-q\rho^{L+2})^2}, \tag{3}$$

where

$$\begin{aligned}
C_1 &= \theta\rho(1+L-2\rho-L\rho+\rho^{L+2}), \\
C_2 &= \lambda(\rho-1)^2.
\end{aligned}$$

In both of the above two cases, the numerators of (2) and (3) are independent of q whereas the denominators of (2) and (3) depend linearly on q and remain positive for all $q \in [0, 1]$. We therefore see that $E\left[W_{[L,q]}\right]$ is either monotonically non-increasing or non-decreasing in $q \in [0, 1]$, given λ, μ, θ and L.

Proposition 2 *Given λ, μ, and θ, there exists $\hat{L}$ such that $E\left[W_{[L,0]}\right] - E\left[W_{[L-1,0]}\right] < 0$ for $L \leq \hat{L}$*

and $E\left[W_{[L,0]}\right] - E\left[W_{[L-1,0]}\right] > 0$ for $L > \hat{L}$.

That is, the response time function decreases in L for $0 \leq L \leq \hat{L}$ and increases in L for $L > \hat{L}$.

Proof: See Appendix B.

From the above two propositions, we easily see that, given λ, μ, and θ, the following algorithm gives the minimum mean system response time and the minimum value of L with $q = 0$ for the threshold parameters:

Starting from $L = 0$, while $E[W_{[L,0]}] \geq E[W_{[L+1,0]}]$, increase L by 1, and otherwise stop. Then the $[L, 0]$ threshold policy brings the minimum mean response time $E[W_{[L,0]}]$.

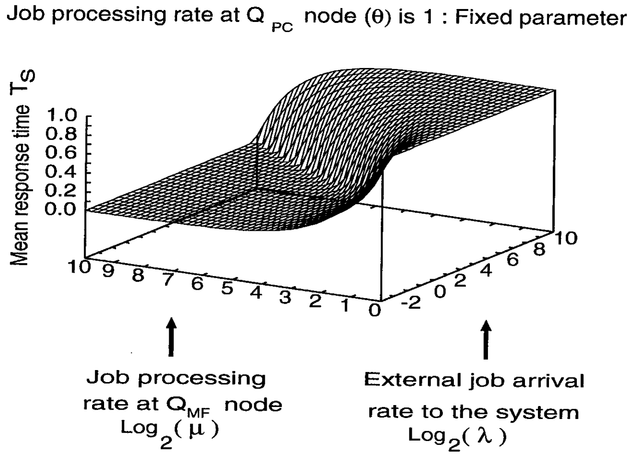

Figure 2: The mean response time T_S by the static optimal policy for each combination of the values of λ and μ.

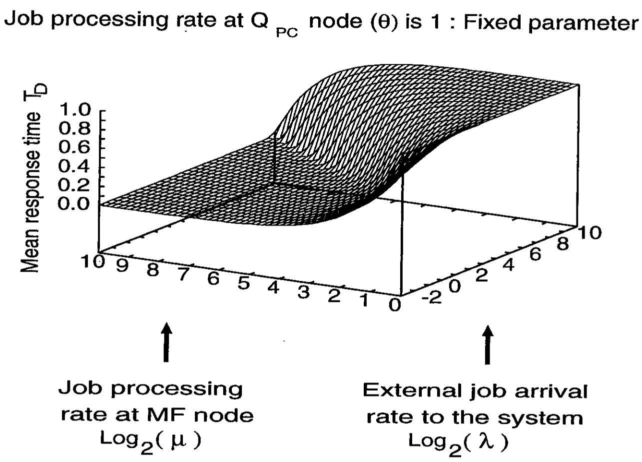

Figure 3: The mean response time T_D by the dynamic optimal policy for each combination of the values of λ and μ.

4 Results and Discussion

We estimate the mean response time of the MF-PC network system for each combination of the values of job arrival rate λ to the system, job processing rate μ at the Q_{MF} node,

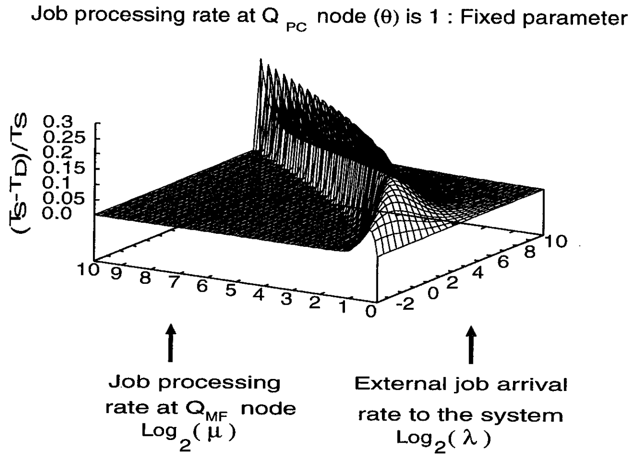

Figure 4: The improvement ratio in the mean response time by the dynamic policy over the static policy for each combination of the values of λ and μ.

and job processing rate θ at the Q_{PC} node. Since we have only three system parameters λ, μ and θ, we scale down θ to 1 and thus we have only two independent parameters. We denote by T_D and T_S, respectively, the mean response times of the dynamic and static policies.

Figures 2 and 3 show the mean response time of the system by the static and dynamic policies, respectively, for various combinations of the values of λ and μ. Define the improvement ratio in the mean response time to be the ratio of the mean response time of the dynamic policy over that of the static policy, i.e., $\frac{T_S - T_D}{T_S}$. Figure 4 shows the improvement ratio in the mean response time with respect to λ and μ. Figure 5 shows, for each given value of λ, the improvement ratio that is maximum with respect to μ The results naturally confirmed our forecast that the dynamic load balancing policy is more effective than the static one. On the other hand, we see that the mean system response time is improved by the optimal dynamic policy over that of the optimal static one at most about 30% in the range of parameter values examined. Note that the difference in the overheads of the two policies are not taken into account. Figure 6 shows the corresponding value of μ that gives the maximum improvement ratio for each value of λ. From this figure, we see that the maximum improvement ratio is achieved for the cases where $\lambda \sim \mu$ for rather large values of both λ and μ.

Another remarkable observation is that if the $[L, q]$ threshold rule is used as the dynamic load balancing policy, the minimum mean system response time is achieved by an $[L, 0]$ threshold rule, that is, the mean system response time can be minimized only by suitably selecting the threshold parameter L and the other threshold parameter q is not effective. Since L is an integer and q whose region is $[0, 1)$ (note that $[L, 1]$ is identical to $[L + 1, 0]$), superficially it might look that the dynamic optimal threshold policy has a continuous parameter $L + q$ to control. The dynamic optimal policy, however, has only the discrete parameter L as the effective parameter to control (see, e.g., Fig. 7) whereas the the optimal static policy has a continuous parameter β_{MF} to control. Three figures, 4, 5 and 6, show seemingly peculiar behaviors concerning the improvement ratio as the values of system parameters change. This peculiarity is thought to come from the contrast between the continuity in the control variable β_{MF} for the static policy and the discreteness in the threshold parameter L for the dynamic policy.

5 Conclusion

We have studied two optimal load balancing policies, static and dynamic, for a system consisting of a single-server central node (Q_{MF}) and an infinite-server satellite node (Q_{PC}) connected by a communication network. By numerical examination, we have estimated the difference in the effects on the mean response time between an optimal dynamic load balancing policy using threshold $[L, q]$ and a static optimal load balancing policy. We have observed that the improvement ratio in the mean response time by the dynamic optimal policy over the static one is at most about 30% in the model examined while overhead due to the policies are not taken into account. The difference is of a certain magnitude for the cases where $\lambda \sim \mu$ for rather large values of both. Another result is that, the minimum mean response time is achieved by the dynamic load balancing policy ($[L, q]$ threshold rule) with threshold parameter $q = 0$ and depending only on the other threshold parameter L.

As the problem in the future, we would like to compare static vs. dynamic individually optimal load balancing policies.

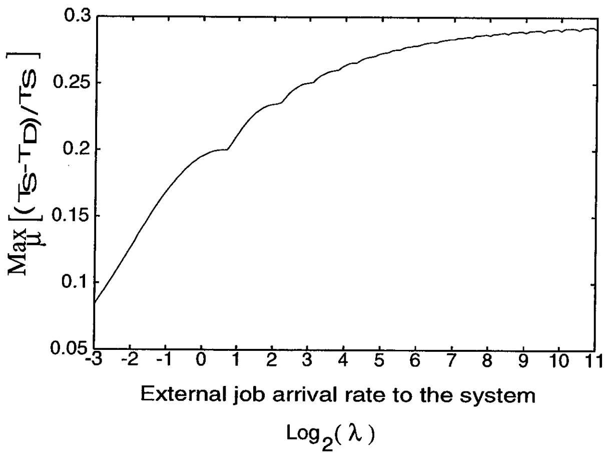

Figure 5: The maximum improvement ratio in the mean response time (with respect to μ) by the dynamic policy over the static policy for each value of λ.

Appendix A: Derivation of $E\left[W_{[L,q]}\right]$

We derive here the mean response time of a job arriving at the system with threshold $[L, q]$, $E\left[W_{[L,q]}\right]$. Let P_k be the probability that the number of jobs in the Q_{MF} node is k. The state transition diagram is shown in Figure 8. With this state transition diagram we

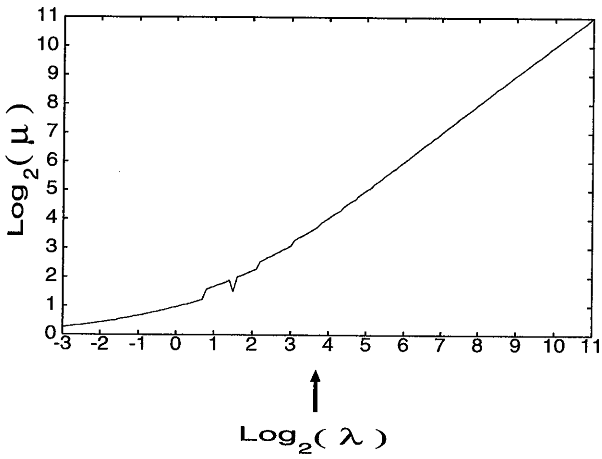

Figure 6: The value of μ that gives the maximum improvement ratio in the mean response time by the dynamic policy over the static policy for each value of λ.

have the following equations:

$$\begin{array}{rcl} \lambda P_0 & = & \mu P_1 \\ \lambda P_1 & = & \mu P_2 \\ \cdots & \cdots & \cdots \\ \lambda P_{L-1} & = & \mu P_L \\ \lambda q P_L & = & \mu P_{L+1}. \end{array} \tag{A.1}$$

Let $\rho = \lambda/\mu$. From (A.1), we can easily have the recursions:

$$\begin{array}{rcl} P_1 & = & \rho P_0 \\ P_2 & = & \rho^2 P_0 \\ \cdots & \cdots & \cdots \\ P_L & = & \rho^L P_0 \\ P_{L+1} & = & \rho^{L+1} q P_0, \end{array} \tag{A.2}$$

and if $\rho = 1$,

$$P_1 = P_2 = \cdots = P_L = P_0, \quad P_{L+1} = qP_0 \tag{A.3}$$

From (A.2), we have

$$\begin{array}{rcl} P_1 + P_2 + \cdots + P_L & = & P_0(\rho + \rho^2 + \cdots + \rho^L) \\ & = & P_0 \dfrac{\rho - \rho^{L+1}}{1-\rho}. \end{array} \tag{A.4}$$

Note that $\sum_{i=0}^{L+1} P_i = 1$. We have

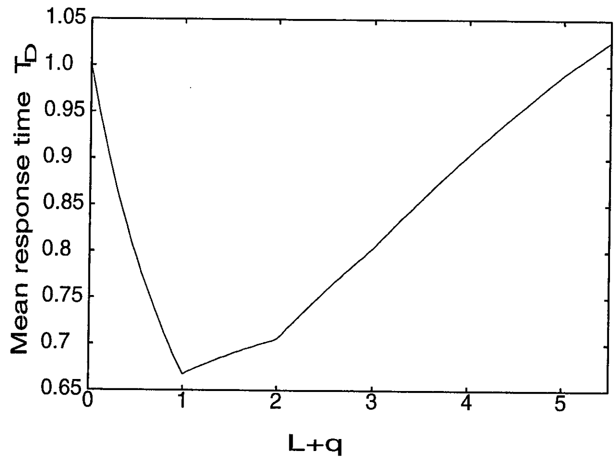

Figure 7: The mean response time by the dynamic policy for each combination of L and q for the case of $\lambda = 1.4142135$ and $\mu = 2.2028464$.

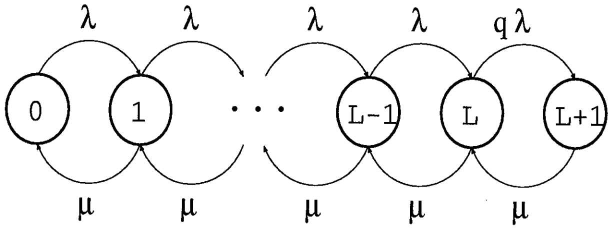

Figure 8: State transition diagram

$$P_0 = \begin{cases} \dfrac{1-\rho}{1-\rho^{L+1}(1-q)-q\rho^{L+2}} & \text{if } \rho \neq 1, \\ \dfrac{1}{L+1+q} & \text{if } \rho = 1. \end{cases} \qquad (A.5)$$

Substituting relation (A.5) to (A.2) or (A.3), we can have the probability that the number of jobs in the Q_{MF} node is k, $P_k(0 \leq k \leq L)$. With the above relations, we proceed to calculate the mean response time of a job arriving at the system. Let P be the probability that a job arriving at the system goes to the Q_{PC} node. With $[L, q]$ threshold rule, the arriving job will go to the Q_{PC} node with probability of 1 if the job finds the Q_{MF} node with states $L+1, L+2, \cdots$, and with probability of $1-q$ if the job finds the Q_{MF} node with state L. Then P is expressed as

$$P = (1-q)P_L + P_{L+1}. \qquad (A.6)$$

The mean response time of a job that goes to Q_{PC} node is θ^{-1}. Let Q be the expected number of jobs (which includes the jobs in service) in the Q_{MF} node from state 0 to state $L+1$ in the state transition diagram. By the Little's Law, the mean response time of a job arriving at the system goes to the Q_{MF} node is

$$QV^{-1},$$

where, V is the actual load rate to the Q_{MF} node, and is given by $V = \lambda(1 - P)$. Therefore, the mean response time of a job arriving at the system with threshold $[L, q]$, $E\left[W_{[L,q]}\right]$, is

$$E\left[W_{[L,q]}\right] = P\theta^{-1} + (1 - P)QV^{-1} = P\theta^{-1} + Q\lambda^{-1}. \tag{A.7}$$

From (A.4), Q can be calculated as follows:

$$Q = \sum_{i=1}^{L} iP_i + (L + 1)P_{L+1}. \tag{A.8}$$

By substituting relations (A.6) and (A.8) into (A.7), we obtain the mean response time of a job arriving at the system with threshold $[L, q]$, $E\left[W_{[L,q]}\right]$. The relation is as follows:

$$E\left[W_{[L,q]}\right] = ((1 - q)P_L + P_{L+1})\theta^{-1} + Q\lambda^{-1}, \tag{A.9}$$

where, if $\rho \neq 1$,

$$\begin{aligned}
P_L &= \rho^L P_0, \\
P_{L+1} &= q\rho^{L+1} P_0, \\
Q &= \sum_{i=1}^{L} iP_i + (L + 1)P_{L+1} \\
&= P_0\rho\frac{(-(L + 1)\rho^L)(1 - \rho) + (1 - \rho^{L+1})}{(1 - \rho)^2} \\
&\quad + (L + 1)P_0 q\rho^{L+1}, \\
P_0 &= \frac{1 - \rho}{1 - \rho^{L+1}(1 - q) - q\rho^{L+2}},
\end{aligned}$$

and if $\rho = 1$,

$$\begin{aligned}
P_L &= P_0, \\
P_{L+1} &= qP_0, \\
Q &= \sum_{i=1}^{L} iP_i + (L + 1)P_{L+1} \\
&= \left(\sum_{i=1}^{L} i + (L + 1)q\right)P_0 \\
&= \left(\frac{L(L + 1)}{2} + (L + 1)q\right)P_0 \\
&= \frac{(L + 1)(L + 2q)}{2(L + 1 + q)}, \\
P_0 &= \frac{1}{L + 1 + q}.
\end{aligned}$$

Appendix B: Proof of Proposition 2

Given λ, μ, θ, and $q = 0$, we have the following two distinct cases:

- Case (1): $\rho = 1$ (i.e., $\lambda = \mu$)
- Case (2): $\rho \neq 1$ (i.e., $\lambda \neq \mu$)

Case (1): $\rho = 1$ (i.e. $\lambda = \mu$)

$$E\left[W_{[L,q]}\right] = \frac{1}{(L+1)}\theta^{-1} + \frac{L}{2}\lambda^{-1},$$

$$\frac{\partial E}{\partial L} = \frac{1}{2\lambda} - \frac{1}{(L+1)^2\theta},$$
$$\frac{\partial^2 E}{\partial L^2} = \frac{2}{(L+1)^3\theta},$$

Thus, $\frac{\partial^2 E}{\partial L^2} \geq 0$ for all values of $L \geq 0$, which means that, the response time function $E\left[W_{[L,q]}\right]$ is convex and hence, it has only one minimum point.

Case (2): $\rho \neq 1$ (i.e. $\lambda \neq \mu$)

$$E\left[W_{[L,q]}\right] = \frac{1}{\mu}\left[\frac{\rho^L}{(1-\rho^{L+1})}[(1-\rho)\frac{\mu}{\theta} - (L+1)] + \frac{1}{(1-\rho)}\right],$$

$$\frac{\partial E}{\partial L} = \frac{\rho^L(\theta(\rho^{L+1}-1) - ((L+1)\theta + (\rho-1)\mu)\log\rho)}{\mu\theta(\rho^{L+1}-1)^2}.$$

Since $\frac{\rho^L}{\mu\theta(\rho^{L+1}-1)^2} > 0$, from the above equation, it is seen that the sign of $\frac{\partial E}{\partial L}$ depends on the value of

$\Delta(L) \triangleq \theta(\rho^{L+1}-1) - [(L+1)\theta + (\rho-1)\mu]\log(\rho)$.

Then, $\frac{d}{dL}\Delta(L) = \theta(\rho^{L+1}-1)\log(\rho)$.

Note that $\Delta(-1) < 0$ Since in this case, $\rho \neq 1$, then we have the following two distinct cases:

- Case (1): $\rho > 1$ By noting that $\log\rho > 0$ and $\rho^{L+1} - 1 > 0$ for $L > -1$, $\frac{d}{dL}\Delta(L) > 0$ for $L > -1$.

- Case (2): $\rho < 1$ By noting that $\log \rho < 0$ and $\rho^{L+1} - 1 < 0$ for $L > -1$, $\frac{d}{dL}\Delta(L) > 0$ for $L > -1$.

We therefore see that $\Delta(L)$ is increasing in L for $L > -1$. Therefore, there exists a unique value $\check{L}$ of L such that $\Delta(\check{L}) = 0$. Note that $\check{L}$ is not necessarily an integer. Thus $E\left[W_{[L,0]}\right]$ decreases with L for $L < \check{L}$ and increases with L for $L > \check{L}$. Note that $[i]$ denotes the largest integer that is not greater than i. Set $\hat{L} = [\check{L}]$ for $\check{L} = [\check{L}]$. For $\check{L} > [\check{L}]$, set $\hat{L} = [\check{L}]$, if $E\left[W_{[\check{L},0]}\right] \leq E\left[W_{[\check{L}+1,0]}\right]$, and $\hat{L} = [\check{L}] + 1$, otherwise. Then, $E\left[W_{[L,0]}\right]$ decreases with L for $L \leq \hat{L}$ and $E\left[W_{[L,0]}\right]$ increases with L for $L > \hat{L}$.

References

[1] E. Altman and N. Shimkin, "Individual equilibrium and learning in processor sharing systems," *Operations Research 46*, pp.776-784, Nov-Dec 1998.

[2] D. L. Eager, E. D. Lazowska, and J. Zahorjan, "Adaptive load sharing in homogeneous distributed systems," *IEEE Trans. Softw. Eng. 12*, 5, pp.662-675, May 1986.

[3] H. Kameda, J. Li, C. Kim and Y. Zhang, *Optimal Load Balancing in Distributed Computer Systems,* Springer, 1997.

[4] C. Kim and H. Kameda, "An algorithm for optimal static load balancing in distributed computer systems," *IEEE Trans. Computers 41*, 3, pp.381-384, 1992.

[5] A. N. Tantawi and D. Towsley, "A general model for optimal static load balancing in star network configurations," In E. Gelenbe, editor, *PERFORMANCE '84*, pp.277-291, Elsevier Science Publishers B. V. (North-Holland), 1984.

[6] A. N. Tantawi and D. Towsley, "Optimal static load balancing in distributed computer systems," *J. ACM 32*, 2, pp.455-465, April 1985.

[7] J. Li and H. Kameda, "A decomposition algorithm for optimal static load balancing in tree hierarchy network configurations," *IEEE Trans. Parallel and Distributed Systems 5*, 5, pp.540-548, 1994.

[8] J. Li and H. Kameda, "Load balancing problems for multiclass jobs in distributed/parallel computer systems," *IEEE Trans. Comput. 47*, 3, pp.322-332, 1998.

[9] Y. Zhang, K. Hakozaki, H. Kameda, and K. Shimizu, "A performance comparison of adaptive and static load balancing in heterogeneous distributed systems," *IEEE 28th Annual Simulation Symposium*, April 9-13. 1995.

[10] Behrooz A. Shirazi, Ali R. Hurson, and Krishna M. Kavi, *Scheduling and Load Balancing in Parallel and Distributed Systems*, IEEE Computer Society press, Los Alamitos, CA, 1995.

[11] S. Pulidas, D. Towsley, and John A. Stankovic, "Imbeding gradient estimators in load balancing algorithms," *Proc. IEEE 8th International Conference on distributed computing systems*, IEEE CS press, pp.482-490, June 1988.

[12] Y. Wang, R. Morris, "Load sharing in distributed systems," *IEEE Trans. on Computers 34*, pp.204-217, March 1985.

[13] R. Mirchandaney, J. Stankovic, "Using stochastic learning automata for job scheduling in distributed processing systems," *J. Parallel and Distributed Computing 3*, pp.527-552, 1986.

[14] J. Stankovic, "Simulations of three adaptive, decentralized controlled, job scheduling algorithms," *Computer Networks 8*, pp. 199-217, 1984.

[15] J. Stankovic, "An application of Bayesian decision theory of decentralized control of job scheduling," *IEEE Trans. on Computers C-34*, 2, pp.117-130, Feb. 1985.

[16] S. Zhou, "A trace-driven simulation study of dynamic load balancing," *IEEE Trans. Soft. Eng. 14*, 9, pp.1327-1341, September 1988.

[17] K.K. Goswami, M.Devarakonda, and R.K. Iyer, "Predication-based dynamic load-sharing heuristics," *IEEE Trans. Parallel and Distributed Systems 4*, 6, pp.638-648,June 1993.

[18] R. Mirchandaney, D. Towsley, and J. A. Stankovic, "Adaptive load sharing in heterogeneous distributed systems," *J. Parallel and Distributed Computing 9*, pp.331-346, 1990.

[19] Kremien, O., and J. Kramer, "Methodical analysis of adaptive load sharing algorithms," *IEEE Trans. Parallel and Distributed Systems 3*, 6, pp. 747-760, Nov. 1992.

[20] Mirchandaney, R., Towsley, D. and Stankovic, J. A. , "Analysis of the effects of delays on load sharing," *IEEE Trans. on Computers 38*, 11, pp.1513-1525, Nov. 1989.

[21] Michael Mitzenmacher, "How useful is old information?" *IEEE Trans. Parallel and Distributed Systems 11*, 1, pp.6-20, January 2000.