한국시뮬레이션학회 논문지
Vol. 24, No. 3, pp. 37-44 (2015. 9)

http://dx.doi.org/10.9709/JKSS.2015.24.3.037
ISSN 1225-5904

Optimal Server Allocation to Parallel Queueing Systems by Computer Simulation

Jin-Won Park*

컴퓨터 시뮬레이션을 이용한 병렬 대기행렬 시스템의 최적 서버 배치 방안

박진원*

ABSTRACT

A queueing system with 2 parallel workstations is common in the field. Typically, the workstations have different features in terms of the inter arrival times of customers and the service times for the customers. Computer simulation study on the optimal server allocation for parallel heterogeneous queueing systems with fixed number of identical servers is presented in this paper. The queueing system is optimized with respect to minimizing the weighted system time of the customers served by 2 parallel workstations. The system time formula for the M/M/c systems in Kendall's notation is known. Thus, we first compute the optimal allocation for parallel M/M/c systems, comparing the results with those from the computer simulation experiments, and have the same results. The CETI rule is devised through optimizing M/M/c cases, which allocates the servers based on Close or Equal Traffic Intensities between workstations. Traffic intensity is defined as the arrival rate divided by the service rate times the number of servers. The CETI rule is shown to work for M/G/c, G/M/c queueing systems by numerous computer simulation experiments, even if the rule cannot be proven analytically. However, the CETI rule is shown not to work for some of G/G/c systems.

Key words : Parallel Queueing systems, Optimal Server allocation, Computer simulation

요 약

실생활에서 2개의 병렬형 대기행렬 시스템은 흔히 발견된다. 병렬형 대기행렬 시스템에서 각 작업장은 서로 다른 고객 도착 패턴과 고객 서비스 시간 분포를 갖는 경우가 많다. 이 논문은 서로 다른 서비스 시간을 갖는 병렬형 대기행렬 시스템에 총 서버 수가 제한되어 있는 상황에서 각각의 작업장에 적절한 수의 서버를 배치하는 문제를 다룬다. 각 작업장은 제한된 수의 서버를 전체 시스템의 가중평균 시스템 시간이 최소가 되는 기준에 따라 배치 받는다. 일반적으로 M/M/c 시스템은 시스템 시간에 대한 해석적 방법의 산출식이 알려져 있다. M/M/c 시스템에 대한 최적 서버 배치 방안을 해석적 방법에 따라 계산한 결과를 컴퓨터 시뮬레이션 실험 결과와 비교해 본 결과, 두 가지 방법에 의한 최적 해가 동일함이 확인되었다. M/M/c 시스템의 최적화 과정에서 발견한, 각 작업장의 유효작업부하가 가장 비슷하거나 같게 되도록 서버를 배치하는 방식인 CETI 규칙에 따라 M/G/c, G/M/c 그리고 G/G/c 시스템에 대한 컴퓨터 시뮬레이션 실험 결과를 제시하였다. 그 결과, 해석적 방법으로는 증명할 수 없지만 일부 G/G/c 시스템을 제외한 나머지 경우에서는 CETI 규칙이 최적의 서버 배치 방식인 것으로 나타났다.

주요어 : 병렬형 대기행렬 시스템, 최적 서버 배치, 컴퓨터 시뮬레이션

*이 논문은 2014학년도 홍익대학교 학술진흥연구비에 의하여 지원되었음.

Received: 10 June 2015, **Revised:** 3 August 2015, **Accepted:** 7 August 2015

***Corresponding Author:** Jin-Won Park
E-mail: jinon@hongik.ac.kr
School of Games, Hongik University

1. Introduction

This paper is concerned with the optimal allocation of servers to parallel queueing systems where the total number of servers are constrained. The servers perform the same working capabilities once they are allocated to a workstation. The study focuses first on the case of 2 parallel workstations with Poisson arrivals and exponential service times where analytic solution is available. Then we extend the study to the cases where arrival and/or service times are non-exponentially distributed with different levels of traffic intensities, ρ.

The problem stems from the allocation of servers in large scale web server cluster systems, where simple web page viewing and knowledge query services are performed separately by a fixed number of web servers[1-2]. With a given number of web severs, the weighted system time of customer's request was examined by computer simulation experiments. This paper is the generalization of the previous study for finding a general rule for allocating servers in an optimal way with respect to minimizing the weighted system time[1-2].

The queueing network models can be applied to various types of working environment, but its numerical analysis is restrictive because of its complexity. Smith et. al.[3] considered optimal server allocation to series, merge, and split topology and their combinations subject to providing a threshold throughput by means of a set of integer number of servers. Their approach focused on allocating servers that can satisfy the given threshold level of throughput. They also listed numerous references concerning the optimal server finite queues and networks. Bitran and Tirupati[4] dealt with the balancing problem for an open network through the reallocation of capacity among the workstations to minimize work-in-process. Dallery and Stecke[5] dealt with the optimal allocation of server and workloads in closed queueing networks. They claimed that the network is balanced with respect to servers if the numer of servers at each station is the same, and is balanced with respect to the workloads if the average workload allocated to each server is the same. Shanthikumar and Yao[6] formulated a nonlinear integer program of allocating servers in a closed queueing network to maximize throughput. They showed that the throughput of the closed queueing network has a monotonic property, such that any optimal allocation must give more servers to stations with a higher workload. Wein[7] proposed a method to determine the service rate (or capacity) that minimize the expected equilibrium customer delay subject to a linear budget constraint on the capacities. Wein first allocated just enough capacity to each station to satisfy its effective arrival rate, and then allocates the excess capacity among the stations in proportion to the sqare roots of their effective arrival rates. Woensel et. al.[8] tried to optimize the number of buffers and servers in a setting of restricted M/G/c/K queueing networks, in the way that the resulting throughput is greater than a predefined threshold throughput. Alexandros et. al.[9] examined the server allocation problem in designing large production lines with reliable multiple identical workstations in series.

Most of the researches surveyed dealt with a closed queueing network, and focused on optimizing server or buffer allocation with respect to minimizing work-in-process or achieving a predefined throughput level. Also, the previous studies focused on the theoretical aspects of a closed queueing networks with respect to the system throughput as a whole. However, this paper concentrates on the server allocation problem for open, parallel but heterogeneous workstations with a fixed number of servers, with respect to maximizing the weighted throughput of the whole system. Besides, our research is concerned with finding a practical rule for optimally allocating servers to parallel workstations even if the rule may not be analytically proven to be optimal. Thus, our research will produce a practically useful optimal server allocation rule, which has not been examined in previous researches.

The paper consists of 4 parts. Following the introduction section, the optimal server allocation problems in functional form and in schematic form are presented. The next section presents the experimental results with exhaustive computer simulation runs under a simple server allocation rule. The final section covers the conclusion and further research issues.

2. The Optimization Problem

A parallel queueing network system consists of two workstations with a number of equally capable servers within a workstation and the total number of servers are fixed. Customers arrive at each workstation separately and are served independently of other workstation. We are concerned with optimally allocating servers to open, parallel but heterogeneous workstations with a fixed number of servers, with respect to maximizing the weighted throughput of the whole system. Consider the following optimization problem which is an analytic model for the problem.

Minimize

$$W=[\lambda_1/(\lambda_1+\lambda_2)]W_1+[\lambda_2/(\lambda_1+\lambda_2)]W_2 \quad (1)$$

subject to

$$c_1+c_2\le c$$

$$0<\lambda_1/c_1\mu_1<1,\ 0<\lambda_2/c_2\mu_2<1, \quad (2)$$

where

$$W_i=1/\lambda_i \bullet [(\lambda_i/\mu_i)^{c_i}\rho_i P_{i0}/c_i!(1-\rho_i)^2]+1/\mu_i,\ i=1,2 \quad (3)$$

$$\rho_i=\lambda_i/(c_i\mu_i),\ i=1,2 \quad (4)$$

and

$$P_{0i}=[\sum_{k=0}^{c_i-1}(\lambda_i/\mu_i)^k/k!+(\lambda_i/\mu_i)^{c_i}/c_i!(1-\rho_i)]^{-1},\ i=1,2. \quad (5)$$

The notations are

W : weighted average system time for workstations 1, 2

W_1, W_2 : average system times for workstations 1, 2

λ_1, λ_2 : arrival rates for workstations 1, 2

c_1, c_2 : numbers of servers for workstations 1, 2

μ_1, μ_2 : service rates for workstations 1, 2.

The schematic diagram for the problem is depicted in Fig. 1, where customers arrive at each workstation with the average arrival rates of λ_1, λ_2, and are served with average rates of μ_1, μ_2. If c_1, c_2 servers are assigned to workstations 1, 2, then the effective service rates will be $c_1\mu_1, c_2\mu_2$ respectively.

The service times, in general, are longer than inter-arrival times, so that a workstation needs a certain number of servers that the effective service rate can exceed the arrival rate. Given a limited number of equally capable servers, allocating the servers to the parallel workstations is searched so that the weighted average waiting time of the customers, W, in the system is minimized.

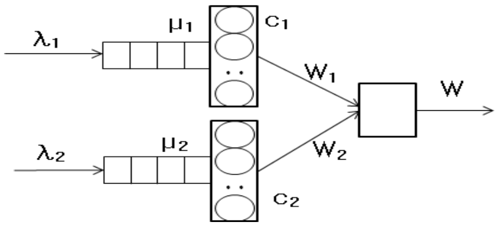

Fig. 1. Schematic diagram for the problem

As shown in equations (1)~(5), W is a function of $\lambda_1, \lambda_2, W_1, W_2$ and W_1, W_2 are the functions of $\lambda_1, \lambda_2, \mu_1, \mu_2$ but their closed forms are complicated even when the arrivals are Poisson fashion and the service times are exponentially distributed.

The problems we are considering were dealt with in Park[2], where hundreds of web servers are allocated for 2 different web services. The web server allocation was optimally performed when the traffic intensities are balanced among the services, called the CETI (Close or Equal Traffic Intensity) rule, which is allocating c_1, c_2 servers in a way that $\lambda_1/c_1\mu_1=\lambda_2/c_2\mu_2$ or as close as possible. This paper is an extension and generalization of the result obtained in Park[1].

The CETI rule is not analytically proven to be optimal but can be shown optimal by computation when the arrival process is Poisson and the service times are exponential. However, if the arrival process is not Poisson or the service time is not exponentially distributed, we may not be able to compute the weighted average system time analytically, but can obtain the weighted average system time by computer simulation experiments. This paper focuses on experimenting whether the CETI rule works with various combinations of inter-arrival time and service time distributions. Table 1 shows the combination of the inter-arrival time distributions and service time distributions that we are considering. In Table 1, the general distributions include Uniform, Triangular and Erlang distributions.

Table 1. Combination of inter-arrival time and service time distributions

Cases	Inter-arrival time distribution	Service time distribution
Case MM	Exponential	Exponential
Case MG	Exponential	General
Case GM	General	Exponential
Case GG	General	General

3. Simulation Experimental Results

The computer simulation experiments are designed in such a way that the effect of allocating the servers to the workstations is maximized. With a small number of total servers, the effect is trivial but with a large number of total servers the effect is indistinguishable. Thus, after many different sets of total number of servers were tested, 10, 14, 18 or 20 sever cases were determined for different set of arrival rates/service rates.

Fig. 2. shows the schematic picture of the model built by ARENA simulation tool. The ARENA model starts with generating customers of 2 different classes, records the randomly generated arrival times and gives services to the customers if servers are available. The model then assigns randomly generated services times, records the departure times when the services are done. At the end of the services, the model computes the total system times of customers. The model itself is simple but needs to be experimented numerous times with various settings of the system parameters.

First, we show the simulation experimental results for Case MM in Table 2. In Case MM, the analytic computation results for the weighted average system times of M/M/c queueing systems can be obtained due to the results described in Lee[10]. The computer simulation experiments last from 1,000 to 11,000 time units, deleting data obtained during the first 1,000 time units. The experiments repeated 5 times typically, but repeated 30 or 50 times when the W(s) values are not distinguishable. Fig. 2. ARENA model for parallel queueing systems.

The notations in Table 2 are the same as in equations (1), (2) and $\rho_1 = \lambda_1/c_1\mu_1$, $\rho_2 = \lambda_2/c_2\mu_2$. Also, W(a), W(s) the weighted average system times from analytic computation and simulation experiments, respectively. The rightmost column in Table 2 shows the accuracy of the simulation results with respect to the analytic computation results, which are close enough to find the optimal allocation of servers. As shown in Table 2, the CETI rule works regardless of the server size, the level of traffic intensities. The results from the analytic computation and from the simulation experiments exactly match in terms of the optimal server allocation.

Table 3 shows the allocation of servers with Case MG, and shows the CETI rule work. As shown in Table 3, it is hard to differentiate the effects of server allocation with light workload cases, where the rightmost column shows the weighted average system times with 5, 30, 50 replications in the second case.

Table 4 shows other MG cases with heavy workloads, where the effects of server allocation is revealed clearly with small number of replications.

Tables 5 and 6 shows the results of Case GM, where the CETI rule works. The simulation experimental results for Case GM are not different from those from Case MG. Similar to Case MG in Table 3, multiple rows in the rightmost columns in Table 5 show the weighted

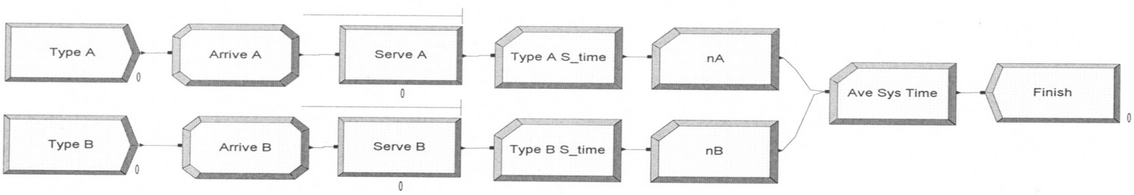

Fig. 2. ARENA Model for parallel queueing Systems

Table 2. Case MM, Poisson Arrival/Exponential Service, C=10, 20

λ_1/λ_2	μ_1/μ_2	(C_1, C_2)	(ρ_1, ρ_2)	W(a)	W(s)	W(s)/W(a)
0.8	0.25	**(4, 6)**	**(0.8000, 0.8000)**	**6.2284 ***	**6.2556 ***	1.0044
1.2	0.25	(5, 5)	(0.6400, 0.9600)	15.0769	14.8798	0.9869
0.4 1.2	0.25 0.25	(2, 8)	(0.8000, 0.6000)	5.9086	5.9414	1.0056
		(3, 7)	**(0.5333, 0.6857)**	**4.5751 ***	**4.6151 ***	1.0087
		(4, 6)	(0.4000, 0.8000)	5.3322	5.3520	1.0037
0.8 1.2	0.15 0.15	(6, 14)	(0.8889, 0.5714)	9.5469	9.5249	0.9977
		(7, 13)	(0.7619, 0.6154)	7.3827	7.3688	0.9981
		(8, 12)	**(0.6667, 0.6667)**	**7.0274 ***	**7.0527 ***	1.0036
		(9, 11)	(0.5926, 0.7273)	7.0744	7.0935	1.0027
		(10, 10)	(0.5333, 0.8000)	7.5151	7.5912	1.0101
0.8 1.2	0.25 0.4	(4, 6)	(0.8000, 0.5000)	4.3424	4.4013	1.0136
		(5, 5)	**(0.6400, 0.6000)**	**3.5336 ***	**3.5379 ***	1.0012
		(6, 4)	(0.5333, 0.7500)	3.9368	3.9627	1.0066
0.2 0.8	0.05 0.15	(6, 14)	(0.6667, 0.3810)	9.9037	9.8400	0.9936
		(7, 13)	(0.5714, 0.4103)	9.5161	9.4750	0.9957
		(8, 12)	(0.5000, 0.4444)	9.4000	9.4302	1.0032
		(9, 11)	**(0.4444, 0.4849)**	**9.3742 ***	**9.3939 ***	1.0021
		(10, 10)	(0.4000, 0.5333)	9.3994	9.4183	1.0020
		(11, 9)	(0.3636, 0.5926)	9.4973	9.5017	1.0005
0.2 0.8	0.03 0.10	(8, 12)	(0.8333, 0.6667)	17.6097	17.9243	1.0179
		(9, 11)	**(0.7407, 0.7273)**	**16.2149 ***	**16.1570 ***	0.9964
		(10, 10)	(0.6667, 0.8000)	16.6526	16.9026	1.0150
		(11, 9)	(0.6061, 0.8889)	20.0349	20.3609	1.0163

Table 3. Case MG, Poisson Arrival/General Service Time, C=20

λ_1/λ_2	Service Time Distribution	(C_1, C_2)	(ρ_1, ρ_2)	W(s)
0.2 0.8	Tria(15, 20, 25) Tria(5, 7, 8)	(7, 13)	(0.5714, 0.4103)	9.4423
		(8, 12)	(0.5000, 0.4444)	9.3551
		(9, 11)*	**(0.4444, 0.4849)**	**9.3479***
		(10, 10)	(0.4000, 0.5333)	9.3669
		(11, 9)	(0.3636, 0.5926)	9.3852
0.2 0.8	Unif(15, 20) Unif(5, 8.3333)	(7, 13)	(0.5714, 0.4103)	9.4028
		(8, 12)	(0.5000, 0.4444)	9.3610
				9.3562
				9.3745
		(9, 11)*	**(0.4444, 0.4849)***	9.3468
				9.3561
				9.3548*
		(10, 10)	(0.4000, 0.5333)	9.3064
				9.3407
				9.3623
		(11. 9)	(0.3636, 0.5926)	9.4393
0.2 0.8	Erlang(10.0, 2) Erlang(3.3333, 2)	(7, 13)	(0.5714, 0.4103)	9.4829
		(8, 12)	(0.5000, 0.4444)	9.4371
		(9, 11)*	**(0.4444, 0.4849)***	**9.3660***
		(10, 10)	(0.4000, 0.5333)	9.4255
		(11. 9)	(0.3636, 0.5926)	9.5143

Table 4. Case MG Poisson Arrival/General Service Time, C=14

λ_1/λ_2	Service Time Distribution	(C_1, C_2)	(ρ_1, ρ_2)	W(s)
0.2 0.8	Tria(15, 20, 25) Tria(5, 7, 8)	(5, 9)	(0.8000, 0.5926)	10.6506
		(6, 8)*	**(0.6667, 0.6667)***	**9.8872***
		(7, 7)	(0.5714, 0.7619)	10.1823
0.2 0.8	Unif(15, 20) Unif(5, 8.3333)	(5, 9)	(0.8000, 0.5926)	10.6658
		(6, 8)*	**(0.6667, 0.6667)***	**9.9729***
		(7, 7)	(0.5714, 0.7619)	10.1889
0.2 0.8	Erlang(10.0, 2) Erlang(3.3333, 2)	(5, 9)	(0.8000, 0.5926)	11.2460
		(6, 8)*	**(0.6667, 0.6667)***	**10.1245***
		(7, 7)	(0.5714, 0.7619)	10.4829

Table 5. Case GM General Arrival/Exponential Service Time, C=18

Service Time Distribution	Service Rates	(C_1, C_2)	(ρ_1, ρ_2)	W(s)
Tria(3, 5, 7) Tria(0.75, 1.25, 1.75)	0.05 0.15	(6, 12)	(0.6667, 0.4444)	9.4968
		(7, 11)	(0.5714, 0.4762)	9.3869
		(8, 10)*	**(0.5000, 0.5333)***	**9.3848***
		(9, 9)	(0.4444, 0.5926)	9.4049
		(10, 8)	(0.4000, 0.6667)	9.4463
Unif(3, 7) Unif(0.75, 1.75)	0.05 0.15	(6, 12)	(0.6667, 0.4444)	9.5626
		(7, 11)	(0.5714, 0.4762)	9.3728
				9.3607
				9.3578
		(8, 10)*	**(0.5000, 0.5333)***	9.3735
				9.3241
				9.3315*
		(9, 9)	(0.4444, 0.5926)	9.3600
				9.3313
				9.3402
		(10, 8)	(0.4000, 0.6667)	9.4291
Erlang(2.5, 2) Erlang(0.625, 2)	0.05 0.15	(6, 12)	(0.6667, 0.4444)	9.6572
		(7, 11)	(0.5714, 0.4762)	9.4829
		(8, 10)*	**(0.5000, 0.5333)***	**9.4131***
		(9, 9)	(0.4444, 0.5926)	9.4789
		(10, 8)	(0.4000, 0.6667)	9.5879

Table 6. Case GM General Arrival/Exponential Service Time, C=14

Service Time Distribution	Service Rates	(C_1, C_2)	(ρ_1, ρ_2)	W(s)
Tria(3, 5, 7) Tria(0.75, 1.25, 1.75)	0.05 0.15	(5, 9)	(0.8000, 0.5926)	10.5308
		(6, 8)*	**(0.6667, 0.6667)***	**9.5520***
		(7, 7)	(0.5714, 0.7619)	9.7224
Unif(3, 7) Unif(0.75, 1.75)	0.05 0.15	(5, 9)	(0.8000, 0.5926)	10.2492
		(6, 8)*	**(0.6667, 0.6667)***	**9.5859***
		(7, 7)	(0.5714, 0.7619)	9.7733
Erlang(2.5, 2) Erlang(0.625, 2)	0.05 0.15	(5, 9)	(0.8000, 0.5926)	10.7302
		(6, 8)*	**(0.6667, 0.6667)***	**10.0337***
		(7, 7)	(0.5714, 0.7619)	10.3254

Table 7. Case GG General Arrival/General Service Time, C=14

Interarrival Time Distribution	Service Time Distribution	(C_1, C_2)	(ρ_1, ρ_2)	W(s)
Erlang(2.5, 2) Erlang(0.625, 2)	Unif(15, 25) Unif(5, 8.3333)	(5, 9)	(0.8000, 0.5926)	9.8785
				9.8342
		(6, 8)*	**(0.6667, 0.6667)***	9.6357
				9.5096*
		(7, 7)	(0.5714, 0.7619)	9.5845
				9.6054
Unif(3, 7) Erlang(0.625, 2)	Tria(15, 20, 25) Unif(5, 8.3333)	**(5, 9)***	**(0.8000, 0.5926)**	9.3968
				9.3742
				9.3746*
		(6, 8)	(0.6667, 0.6667)*	9.4066
				9.4060
				9.4049
		(7, 7)	(0.5714, 0.7619)	9.5949
				9.5902
				9.5924
Tria(3, 5, 7) Unif(0.75, 1.75)	Erlang(10, 2) Tria(5, 7, 8)	(5, 9)	(0.8000, 0.5926)	9.6899
		(6, 8)	(0.6667, 0.6667)*	9.3738
				9.3850
				9.3916
		(7, 7)*	**(0.5714, 0.7619)***	9.3326
				9.3351
				9.3356*
		(8, 6)	(0.5000, 0.8889)	9.4443
Erlang(2.5, 2) Tria(0.75, 1.25, 1.75)	Unif(15, 25) Erlang(3.3333, 2)	(5, 9)	(0.8000, 0.5926)	9.7080
		(6, 8)*	**(0.6667, 0.6667)***	**9.4390***
		(7, 7)	(0.5714, 0.7619)	9.5073

average system times with 5, 30 and 50 replications. With 50 replications, the allocation (8, 10) shows the optimal for the case.

Table 7 shows the results for Case GG, where the inter-arrival time distributions are not exponential and the service time distributions are not exponential either.

The rightmost columns in Table 7 show the weighted average system times with 5 and 30 replications for the first case, and with 5, 30 and 50 replications for the second and the third cases. The simulation experiments are extended until the 95% confidence intervals are not overlapped. Quite different from the conjecture that the CETI rule would also work in Case GG, the CETI rule does not work in some of GG cases.

From the computer simulation experiments presented in this section, the CETI rule may work with M/M/c, M/G/c and G/M/c cases where either inter-arrival time or service time distributions are exponential, but does not always work when both are not exponential.

4. Conclusion and Further Research Issues

This paper is concerned with the optimal allocation of servers to parallel queueing networks where the total number of servers is constrained. The study focuses first on the 2 parallel queueing systems with Poisson arrivals and exponential service times where analytic solution is available. Then the non-exponential cases of inter-arrival time and/or service time distributions were also studied.

From the numerous computer simulation experiments, the CETI rule, allocating servers to the parallel workstations in a way that both workstations having close or equal traffic intensities, may work with M/M/c, M/G/c and G/M/c cases but not with some of G/G/c cases, where M means exponential and G means general distributions. However, the CETI rule remains only a conjecture since the rule cannot be proven analytically.

The result obtained from our research may be applied to many different real world situations. The result of our research is simple enough to apply to many different types of parallel workstations in allocating limited number of servers. For example, banks may allocate tellers to general services and loan services, hospitals may allocate doctors to emergency services and general services and web server managers may allocate web servers to web page search services and query services. In handling above mentioned problems, the CETI rule may be useful. The more comprehensive and exhaustive simulation experiments may be required to show the robustness of the CETI rule, or to disprove the CETI rule.

References

1. Park, Jin-Won, Operational Scheme for large Scale Web Server Cluster Systems, J. of the Korea Society for Simulation, Vol. 22, No. 3, pp. 71-79, in Korean, 2013.
2. Park, Jin-Won, Analysis on the Performance Elements of Web Server Cluster Systems, J. of the Korea Society for Simulation, Vol. 19, No. 3, pp. 91-98, in Korean, 2010.
3. Smith JM, Cruz FR, Woensel TV, “Optimal Sever Allocation in General, Finite, Multi-server Queueing Networks,” Applied Stochastic Models in Business and Industry, 26, pp. 705-736, 2010.
4. Bitran GR, Tirupati D, “Tradeoff Courves, Targeting and Balancing in Manufacturing Queueing Networks,” Operations Research, Vol. 37, No. 4, July-August, 1989.
5. Dallery Y, Stecke KE, “On the Optimal Allocation of Servers and Workloads in Closed Queueing Networks,” Operations Research, Vol. 38, No. 4, July-August, 1990.
6. Shanthikumar JG, Yao DD, “On Server Allocation in Multiple Center Manufacturing Systems,” Wein LM, “Capacity Allocation in Generalized Jackson Networks,” Operations ResearchVol. 36, No. 2, March-April, 1988.
7. Wein LM, “Capacity Allocation in Generalized Jackson Networks,” Operations Research Letters 8, pp. 143-146, 1989.
8. Woensel TV, Andriansyah R, Cruz FR, Smith JM, Kerbache L, “Buffer and Server Allocation in General Multi-Server Queueing Networks,” International Transactions in Opertations Research, Vol. 17, No. 2, pp. 257-286, 2010.
9. Alexandros DC, Chrissoleon PT, “On the Server Allocation in Large Reliable Production Lines with Exponential Process Times,” In 5th Internation Conference on “Analysis of Manufacturing Systems-Production Management,” Zakynthos Island, Greece, 2005.
10. Lee, Howoo, Theory of Queueing Systems, Sigma Press, in Korean, 1998.

박 진 원 (jinon@hongik.ac.kr)

1975　서울대학교 공과대학 산업공학과 학사
1987　미국 오하이오 주립대학교 산업시스템공학과 석사, 박사
1988～1999　한국전자통신연구원 책임연구원
2000～현재　홍익대학교 게임소프트웨어전공 교수

관심분야 : 모델링&시뮬레이션, 시뮬레이션을 이용한 시스템 최적화, 공학교육