Join-The-Shortest Queue Policy In Web Server Farms

GJCST Classification
E.1. C.2.4

Satheesh Abimannan[1] Kumar Durai[2] A.V.Jeyakumar[3] Krishnaveni.S[4]

***Abstract*-In the web server farm, the Join the Shortest Queue (JSQ) routing policy is well-liked. This policy is optimal in single-server queues system. But it is very difficult to analyze in multiple server system. The web server farm consists of N identical queues with infinite buffers, and each of the queue has one server. When a job arrives at the system, it is sent to the queue with smallest number of jobs. For exponential multi server systems with queue in parallel in which jobs are enter into one of the shortest queue upon arrival and in which jockeying is not possible. The objective of this paper is to compute the possibility of worst case for systems in which the new arrival job join one of the shortest queues upon arrival. We used the modified power-series algorithm to compute the stationary queue length.**

Keywords-Join-the-shortest queue, multi server system, parallel queues, response time, No jockeying, power-series algorithm.

I. Introduction

The server farm is a popular architecture of computing centers. It consists of a front-end router/dispatcher which receives all the incoming requests (jobs), and dispatches each job to one of a collection of servers which do the actual processing, as depicted in Figure 1. The dispatcher employs scalability (it is easy to add and remove servers) and high reliability (failure of individual servers does not bring the whole system down). One of the most important design goals of a server farm is choosing a routing policy which a routing policy (also called a "task assignment policy", or TAP), which decides when and to which server an incoming request should be routed. Server farms afford low cost (many slow servers are cheaper than one fast server), high will yield low response times; the response time is the time from the arrival of a request to its completion.

In this paper we consider web server farm architecture serving static request. Requests for files (or HTTP pages) arrive at a front-end dispatcher. The dispatcher then immediately routes the request to one of the servers in the farm for processing using a JSQ routing policy. It is important that the dispatcher does not hold back the arriving connection request or the client will time out and possibly submit more requests

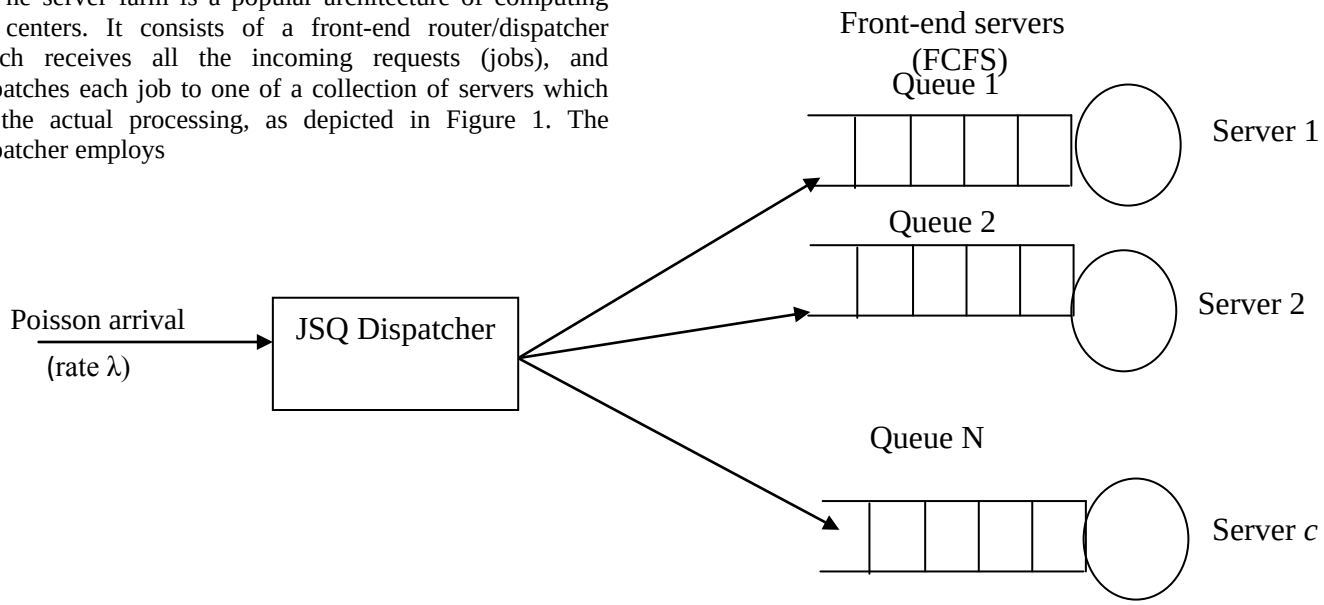

Figure 1. Server farm with front-end dispatcher and K identical FCFS back-end servers

About[1]Department of Computer Science and Engineering,
About[2]Department of Electronics and Communication Engineering
About[3]Department of Mathematics,
About[4]Department of Computer Applications
**Periyar Maniammai University, Vallam, India*

A. Model and Notation

The system considered in this paper consists of N, N≥2, identical queues each of which has buffer with infinite capacity; each of the queues has single servers. Each queue is served in a First-come-First-service (FCFS) order. Let *c* be the number of servers in the system. A job dispatcher is used to assign jobs to queues. The job arrival process to the

system is assumed to be Poisson with rate λ. The job service times are assumed to be exponentially distributed with mean 1/μ. At the arrival instant, a job is sent to one of the queues according to the join the Shortest Queue (JSQ) policy; i.e., it is assigned to the queue with the smallest number of jobs. No jockeying between queues is permitted.

B. *Contribution/Outline*

In a web server farms it has $c \geq 2$ parallel server. Services performed by server j have an exponentially distributed duration with a mean $1/\mu_j$, j = 1,…,*c*. While job arrives at web server farms it will immediately assign any one of the server randomly and assign the queue which has minimal job size. The arrival job enter in such systems often notice that the job in other queues are being served faster than those in their own queue, and that they are overtaken by job that arrived later. Of course, this phenomenon may be due to different skills, and hence different service rates, among the servers. But even if the service rates of all servers are equal, this phenomenon frequently occurs. A simple explanation is found by considering the situation that the arrival job meets an equal number of jobs in the system $n \geq 1$ in each of the queues upon arrival. Then, by the lack of memory of the exponential service time distributions and the symmetry of the system, each queue has the same possibility of becoming the queue that is soonest exempted of its *n* jobs. Hence, the arriving job has in this situation *(c - 1)/c* of chances that the job does not join the queue in which his service would have started earliest.

II. Prior work

The systems with single server queues, the JSQ policy has been proved to be optimal in that it maximizes the throughput of the system and also minimizes the expected total time to complete the service of all jobs arriving before some fixed time. For the case N = 2, Height [9] studied the JSQ problem allowing jockeying between two queues. Zhao and Grossman[15] developed an algorithm for computing the probability that are exactly k jobs in each queue and then finding the joint distribution of the queue lengths in the system. The matrix-geometric approach, as introduced by M.F.Neuts [12] in his book, has proved to be powerful tool for the analysis of Markov processes with large and complicated state spaces, particularly the ones that appear when modeling Queueing or maintenance systems. Gertsbakh and Kao et al. [8][10][13] used the matrix-geometric technique to calculate the state occupancy probabilities approximately for two-queue systems with unequal service rates. Adan, Wessels and Zijm [1], using one partitioning of the state space, obtained an explicit ergodicity condition from Neuts' mean drift condition and also explicitly determined another partitioning of the associated R-matrix. [6] In his paper in IEEE Transaction on computers appeared in 1990, F.Bonomi compared the job assignment problem with processor sharing queues in the JSQ policies with First-come-First (FCFS) service. He demonstrate that the JSQ policy offers a very good solution to the job assignment problem for PS parallel system, even though this is not necessarily optimal for nonexponential service time distribution. In 1996, Lin and C.S.Raghvandra [11] developed a method to analyze the performance of the JSQ policy, applicable for systems with both single server and multiserver queues, assuming the job arrival process to be Poisson and service time distribution exponential. This method uses birth-death markov process to model the evaluation of the number of jobs in the system using simulation. Later, Harchol Balter et.al., [14] provided the first approximate analysis of JSQ in the PS server farm model for general job size distributions and obtained the distribution of queue length at each queue. For this, they approximate queue length of each queue in the server farm by a one dimensional Markov chain. The aim of the present paper is to compute the possibility of worst case for systems in which the job join one of the shortest queues upon arrival. For the computations reported in this paper we have used the modified power-series algorithm to compute the stationary queue length distribution as described in Blanc [2],[3],[4],[5] for the shortest-queue system.

The rest of this paper is organized as follows. The analytical model of the system is discussed in section 3. Finally, some concluding remarks are made in section 4.

III. Analytical model

A. *Homogeneous servers*

In the first case we consider the servers in the web server farms are homogeneous, which is the service rate of all servers are equal, μj = μ, j = 1,…,c, and the arrival job joins one of the shortest queues with equal possibilities. The system load can be defined as ρ = λ / (Ncμ), and for stability it is assumed that ρ < 1.Given that the arrival job joins a queue in which n jobs were already present, the waiting time Wn of this new arrival job as an Erlang distribution with mean n/μ and consist of n phases, n = 1,2,…, by the assumption of exponential service times. The other possibility of the arrival job join the another queue is defined as follows. Suppose the system is in state (n1,…,nc), with nk the length of the queue k, k = 1,…,c, and the arriving job join the queue j, the ϕj(n1,…,nc) is the possibility of that some other server i, $i \neq j$, will be the first to complete service of its current ni jobs. This probability can be determined from relation

$$\phi_j(n_1, \dots, n_c) = Pr\left\{\min_{i=1,\dots c} W_{n_i} < W_{n_j}\right\}, j = 1, \dots, c; \quad (1)$$

Here, Wni, i =1,…, c, represent independent, Erlang distributed random variable with mean ni/μ and consisting of ni phases. To keep notation simple this probability will be evaluated for the case j=1; the other cases follow by interchanging the indices. Clearly, if n1=0 an arriving job has zero waiting time, and, hence, for all n2,…,nc ϵ IN,

ϕ1(0,n2,…,nc) = 0 (2)

Next, let n1 ≥ 1. By conditioning on the length y of the n1 services in queue 1 this conditional probability becomes, for n2,…,nc ≥ 1,

$$\phi_1(n_1,\dots,n_c) = 1 - \int_0^\infty Pr\{W_{n_2} > y,\dots,W_{n_c} > y\} d\,\Pr\{W_{n_1} \le y\} \quad (3)$$

By the independence of the service by the various servers this can be written as

$$\phi_1(n_1,\dots,n_c) = 1 - \int_0^\infty Pr\{W_{n_2} > y\}\dots\Pr\{W_{n_c} > y\} d\,\Pr\{W_{n_1} \le y\} \quad (4)$$

Using the explicit expression for the Erlang distribution and its follows that

$$\phi_1(n_1,\dots,n_c) = 1 - \int_0^\infty \left[\prod_{j=2}^{c} \sum_{i_j=0}^{nj-1} \frac{(\lambda y)^{i_j}}{i_j!} e^{-\lambda y}\right] \lambda \frac{(\lambda y)^{n_1-1}}{(n_1-1)!} e^{-\lambda y} dy \quad (5)$$

By interchanging the order of summation and integration this expression can be written as

$$\phi_1(n_1,\dots,n_c) = 1 - \sum_{i_2=0}^{n_2-1} \dots \sum_{i_c=0}^{n_c-1} \frac{1}{(n_1-1)!2!\dots i_c!} \int_0^\infty \lambda(\lambda y)^{n_1+i_2+\dots i_c-1} e^{-c\lambda y} dy \quad (6)$$

This integral can be evaluated as, for $n_1,\dots,n_c \ge 1$,

$$\phi_1(n_1,\dots,n_c) = 1 - \sum_{i_2=0}^{n_2-1} \dots \sum_{i_c=0}^{n_c-1} \frac{(n_1+i_2+\dots+i_c-1)!}{(n_1-1)!2!\dots i_c!} \frac{1}{c^{n_1+i_2+\dots+i_c}} \quad (7)$$

In the special case that all queues are equally short this probability becomes. For $n \ge 1$,

$$\phi_1(n_1,\dots,n_c) = 1 - \sum_{i_2=0}^{n_2-1} \dots \sum_{i_c=0}^{n_c-1} \frac{(n_1+i_2+\dots+i_c-1)!}{(n_1-1)!2!\dots i_c!} \frac{1}{c^{n_1+i_2+\dots+i_c}} = 1 - \frac{1}{c} = \frac{c-1}{c}, \quad (8)$$

which is immediate for homogeneous system, as mention in section 3.1.

Table 1: Worst case for joining the new arrival job in queue 1 in the homogeneous system with c=2

n_2/n_1	**1**	**2**	**3**	**4**	**5**	**6**
6	0.0156	0.0625	0.1445	0.2539	0.3770	0.5000
5	0.0313	0.1094	0.2266	0.3633	0.5000	0.6230
4	0.0625	0.1875	0.3438	0.5000	0.6367	0.7461
3	0.1250	0.3125	0.5000	0.6563	0.7734	0.8555
2	0.2500	0.5000	0.6875	0.8125	0.8906	0.9375
1	0.5000	0.7500	0.8750	0.9375	0.9688	0.9844

able 2: Worst case for joining the new arrival job in queue 1

if n_1 = 2 in the homogeneous system with c=3

n_3/n_2	**2**	**3**	**4**	**5**	**6**
6	0.5066	0.3271	0.2117	0.1431	0.1045
5	0.5158	0.3448	0.2379	0.1764	0.1431
4	0.5364	0.3813	0.2887	0.2379	0.2117
3	0.5802	0.4527	0.3813	0.3448	0.3271
2	0.6667	0.5802	0.5364	0.5158	0.5066

Table 1 shows the worst case of $\phi_1 (n_1, n_2)$ for new arrival job joining queue 1 in the case $c = 2$, for $n_1, n_2 = 1,\ldots,6$. Note that the values $\phi_1 (n + m, n), n \geq 1, m \geq 1$, are irrelevant since an arriving job will join the shorter queue, and, hence, not queue 1 in these states. Further, observe that $\phi_1 (n, n+m) \rightarrow 0$ as $m \rightarrow \infty$ for fixed $n \geq 1$, but that $\phi_1 (n , n + m)$ increases with increasing n for fixed $m \geq 1$. Moreover, using (7) it follows with the aid of Stirling's formula that for fixed $m \geq 1$, as $n \rightarrow \infty$,

$$\phi_1 (n, n + m) = 1 - \sum_{i=0}^{n+m-1} \frac{(n+i-1)!}{(n-1)!i!} \frac{1}{2^{n+i}} = \frac{1}{2} - \sum_{k=0}^{m-1} \binom{2n+k-1}{n-1} \frac{1}{2^{2n+k}} \uparrow \frac{1}{2} \quad (9)$$

Table 2 shows the worst case of $\phi_1 (2, n_2 , n_3)$ for arriving job joining queue 1 in the case c=3, for n_2, $n_3 = 2,\ldots,6$. Note that $\phi_1 (2, 2+m, 2) \rightarrow \frac{1}{2}$ as $m \rightarrow \infty$, which agrees with values of $\phi_1 (2, 2)$ for $c = 2$. More generally, as $m \rightarrow \infty$, $\phi_1 (n, n+k+m) = \phi_1 (n, n+k+m, n + k)$ tends to the value of $\phi_1 (n, n+k)$ for $c=2$. For instance, for $n = 2$ and $k = 1$ the limit is $\phi_1 (2, 3)=0.3125$, see Tables 2 and 1. Hence, the limiting behavior of the conditional probabilities for $c = 3$ is more complex than that for $c = 2$. However, the most important property is that parallel to the main diagonal $n_1 = n_2 = n_3$ these probabilities tend to $\frac{2}{3}$, although rather slowly. For instance, $\phi_1 (n, n, n+1) = \phi_1 (n, n+1, n)$ equals 0.6527 for $n = 100$ and 0.6568 for $n = 200$, while $\phi_1 (n, n+1, n+1)$equals 0.6379 for $n = 100$ and 0.6464 for $n = 200$.

The (unconditional) probability of worst case is defined as

$$P_{BL} = \sum_{n_1=1}^{\infty} \ldots \sum_{n_c=1}^{\infty} p(n_1 , \ldots, n_c) \sum_{j=1}^{c} \Upsilon_j (n_1, \ldots n_c) \phi_j (n_1, \ldots, n_c); \quad (10)$$

here, $\Upsilon_j (n_1, \ldots, n_c)$, $j= 1,\ldots,c,$ denotes the probability that a job joins queue j when the system is in state $(n_1,\ldots,n_c)$. It is defined by, with $I_{\{.\}}$ the indicator function,

$$\Upsilon_j (n_1, \ldots, n_c) = I_{\{\forall i\, n_i \geq n_j\}} / \sum_{i=1}^{c} I_{\{n_i = n_j\}}, \; j = 1, \ldots, c, n_1, \ldots, n_c \in \mathbf{IN}; \quad (11)$$

in particular, $\Upsilon_j (n_1, \ldots, n_c) = 0$ whenever $n_j > n_i$ for some $i \neq j, j = 1, \ldots, c$. For application of the power-series algorithm, the stability state of probabilities $p(n_1, \ldots, n_c)$ of the joint queue length process in equation (10) are represented as

$$p(n_1, \dots, n_c) = \rho^{n_1+\cdots+n_c} \sum_{k=0}^{\infty} \rho^k \, b(k; n_1, \dots, n_c), n_1, \dots, n_c \in \mathbf{IN}. \quad (12)$$

The coefficients $b(k; n_1, \dots n_c)$ can be recursively computed by scheme (see Blanc 1987a, 1987b, 1992) that follows after substitution of equation (12) into the following global balance equation

$$\left[\lambda + \sum_{j=1}^{c} \gamma_j I_{\{n_j \geq 1\}}\right] p(n) = \lambda \sum_{j=1}^{c} \gamma_j (n - e_j) I_{\{n_j \geq 1\}} p(n - e_j) + \sum_{j=1}^{c} \gamma_j \, p(n + e_j); \quad (13)$$

Here, $n = (n_1, \dots, n_c) \in \mathbf{IN}^c$ denotes a state vector, and e_j are vector of all zeros except a 1 at the jth coordinate, $j = 1, \dots, c$.

Figure 2 shows the possibility of Worst case for to join the job in a queue of homogeneous servers with c = 2,3,4,5 servers, respectively, and a fixed service capacity of $c\gamma = 1$, as a function of the load ρ. Recall that $\rho = \lambda < 1$ if $c\mu = 1$. It can be seen that at fixed, low values of ρ the probability of Worst case is decreasing with the number of servers. This can be explained by noting that in light traffic the possibility that a new arrival job finds an idle server upon arrival, and hence has zero probability of worst case, increases with an increasing number of servers. In fact, it follows from the power-series expansion at ρ = 0 that in light traffic: for c = 2,3,…,

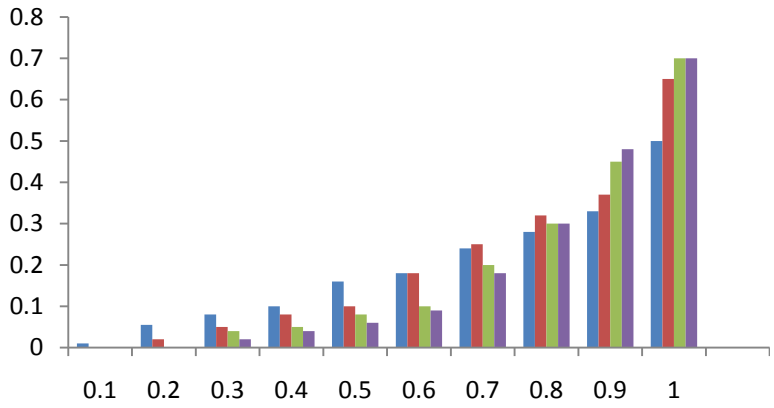

Figure 2: Probability of Worst case in homogeneous systems, for c = 2,3,4,5.

$$P_{BL} \sim \frac{c^{c-2}\rho^c}{(c-2)!} - \frac{c^{c-2}\rho^{c+1}}{c!} (c^3 - c^2 - c + 2) + O(\rho^{c+2}), \rho \downarrow 0. \quad (14)$$

On the other hand, the figure 2 shows that at fixed values of ρ close to 1 the possibility of worst-case increasing with the number of servers. For these moderate number of servers the possibility of worst-case seems to tend to $(c-1)/c$ as $\rho \to 1$. This is supported by (9) for the case c=2.

3.2 Heterogeneous servers

In the second case we consider a heterogeneous system in which server j servers request at γ_{j} ,j =1,…,c. The arrival jobs are supposed to be not aware of these differences among the servers, and still join the shortest queue upon arrival. Hence, we will apply (11) unless stated otherwise. Expression (2.7) is generalized for this case, for $n_1, \dots, n_c \geq 1$,

$$\phi_1(n_1, \dots, n_c) = 1 - \sum_{i_2=0}^{n_2-1} \dots \sum_{i_c=0}^{n_c-1} \frac{(n_1+i_2+\cdots+i_c-1)!}{(n_1-1)! i_2! \dots i_c!} \frac{\gamma_1^{n_1} \gamma_2^{i_2} \gamma_c^{i_c}}{(\gamma_1+\cdots+\gamma_c)^{n_1+i_2+\cdots+i_c}}. \quad (15)$$

Table 3: Conditional probability of Worst-Case if queue 1 is joined, for $c = 2,\ \mu_1 = 1.2,\ \ \mu_2 = 0.8$

n_2/n_1	1	2	3	4	5	6
6	0.0041	0.0188	0.0498	0.0994	0.1662	0.2465
5	0.0102	0.0410	0.0963	0.1737	0.2666	0.3669
4	0.0256	0.0870	0.1792	0.2898	0.4059	0.5174
3	0.0640	0.1792	0.3174	0.4557	0.5801	0.6846
2	0.1600	0.3520	0.5248	0.6630	0.7667	0.8414
1	0.4000	0.6400	0.7840	0.8704	0.9222	0.9533

Table 3 shows the conditional probability of worst-case $\phi_1(n_1, n_2)$for the arrival job joining queue 1 in the case $c = 2, \mu_1 = 1.2, \mu_2 = 0.8$ for $n_1, n_2 = 1, \ldots, 6$. The values $\phi_1(n + m, n), n \geq 1, m \geq 1$, are again irrelevant as in Table 1, but they indicate that in some cases (when $\phi_1(n + m, n) \leq \frac{1}{2}$) arriving jobs would be better off if they did not join the shorter queue. Further, note that $\phi_2(n_1, n_2) = 1 - \phi_1(n_1, n_2)$for all $n_1, n_2 = 1,2, \ldots$.

In lightly to moderately loaded systems, heterogeneous in the service rates increases the probability of worst case. This has more to do with an increase of congestion with increasing difference between the service rates than with the conditional probabilities of worst-case. For instance, $P_{BL} \sim p(1,1)\left[\frac{1}{2}\phi_1(1,1) + \frac{1}{2}\phi_2(1,1)\right] (\rho \downarrow 0)$, see (10), (12) and

$p(1,1) \sim \frac{1}{2}\rho^2 \frac{(\mu_1+\mu_2)^2}{\mu_1\mu_2}$ $(\rho \downarrow 0)$ increases for fixed (small) load ρ as $\mu_1 = 2 - \mu_2$ increases, while $\frac{1}{2}\phi_1(1,1) + \frac{1}{2}\phi_2(1,1) = \frac{1}{2}$ remains constant.

Suppose the system is heavily loaded, in the heterogeneous servers service rates decreases the possibility of worst-case. This can be explained by the features that if server 1 works faster ($\mu_1 > \mu_2$), the joint queue length process will tend to spend more time in the area $n_1 < n_2$ than in the area $n_1 > n_2$, while for $n_1 < n_2$, $\phi_1(n_1, n_2)$ is smaller than its opposite $\phi_2(n_1, n_2) = 1 - \phi_1(n_2, n_1)$, see Table 3. A further analysis indicates that PBL approaches $\frac{\mu_2}{(\mu_1+\mu_2)}$ as $\rho \uparrow 1$ if $\mu_1 > \mu_2$, while the approach of this limit is less steep with increasing value of $\mu_1 = 2 - \mu_2, 1 \leq \mu_1 \leq 2$. This limit is obtained from numerical analysis. There is no simple generalization of (9) to the heterogeneous system, since, e.g., $\phi_1(n, n)\downarrow 0$ as $n \to \infty$, see Table 3.

IV. Conclusion

This paper has studied the analysis of worst-case in JSQ routing policy in web server farms. A new arrival job is said to experience bad luck (Worst-case) if it joined one of the shortest queues upon arrival, but it service would have started earlier if it had joined one of the other queues. In homogeneous system, the possibility of worst case may well exceed $\frac{1}{2}$ when there are three or more servers, but this only occurs if the load of the system is very close to 1. The approach of this probability to its heavy traffic limit is very steep, so that this limit, which is easily computable, will not be a good approximation for most values of the load. Heterogeneous in the service rates tends to increase this probability in light traffic, but to decrease it in moderate to heavy traffic.

V. References

1) Adan.I.J.B.F, J.Wessels, W.H.M. Zijm, Matrix-geometric analysis of the shortest queue problem with threshold jockeying, Oper.Res.Lett.13 (1993) 107-112.
2) Blanc, J.P.C., . A note on waiting times in systems with queues in parallel, Journal of Applied Probability 24 (1987): 540–546.
3) Blanc, J.P.C., . On a numerical method for calculating state probabilities for queueing systems with more than one waiting line, *Journal of Computational and Applied Mathematics* 20 (1987): 119–125.
4) Blanc, J.P.C.,. The power-series algorithm applied to the shortest-queue model, *Operations Research* 40(1992): 157–167.
5) Blanc, J.P.C.,. The power-series algorithm applied to the shortest-queue model, *Operations Research* 40(1992): 157–167.
6) Bonomi.F, On job assignment for a parallel system of processor sharing queues, IEEE Trans. Comput. 39 (7) (1990) 858-869.
7) Donald Gross, Carl M.Harris, Fundamentals of Queueing Theory, Third Edition, John Wiley & Sons, Inc 2004.
8) Gertsbakh.I, The shorter queue problem: A numerical study using the matrix geometric solution, European J. Oper.res.15, 374-381 (1984).
9) Haight.F.A, Two Queues in Parallel, Biometrika, vol.45,pp.401-410,1958.

10) Kao.E.P.C and C.Lin, A matrix-geometric solution of the jockeying problem, European J.Oper.Res.44, 67-74 (1990).
11) Lin.H.C, C.S.Raghavendra, An analysis of the join the shortest queue (JSQ) policy, IEEE Trans. Parallel and Distributed systems, 7(1996) 301-307.
12) Neuts.M.F,Matrix-geometric solutions in stochastic models, Johns Hopkins University Press, Baltimore, MD 1981.
13) Ramaswami.V and G. Latouche, A general class of Markov processes with explicit matrix-geometric solutions, OR Spektrum 8, 209-218 (1986).
14) Varun Gupta, Mor Harchol Balter, Karl Sigman, Ward Whitt, Analysis of Join-the-shortest-queue routing for web server farms, Performance Evaluation 64 (2007) 1062-1081.
15) Zhao.Y and W.K.Grassmann, A Numerically Stable Algorithm for Two Server Queue Models, Queueeing Systems, vol.8,pp.59-79,1991.