The Asymptotics of Selecting the Shortest of Two, Improved

Michael Mitzenmacher* Berhold Vöcking †

Abstract

We investigate variations of a novel, recently proposed load balancing scheme based on small amounts of choice. The static (hashing) setting is modeled as a balls-and-bins process. The balls are sequentially placed into bins, with each ball selecting d bins randomly and going to the bin with the fewest balls. A similar dynamic setting is modeled as a scenario where tasks arrive as a Poisson process at a bank of FIFO servers and queue at one for service. Tasks probe a small random sample of servers in the bank and queue at the server with the fewest tasks.

Recently it has been shown that breaking ties in a fixed, asymmetric fashion, actually improves performance, whereas in all previous analyses, ties were broken randomly. We demonstrate the nature of this improvement using fluid limit models, suggest further improvements, and verify and quantify the improvement through simulations.

1 Introduction

In this paper, we study a novel load balancing scheme proposed by Vöcking [13] using fluid limit models. We study his scheme both in the static scenario, which corresponds to a hashing scheme using multiple hash functions and can be described as distributing a fixed number of balls into a fixed number of bins, and a natural queueing setting, where tasks arrive at a bank of servers and queue at one for service.

*Harvard University, Computer Science Department. 29 Oxford St., Cambridge, MA 02138. e-mail:michaelm@eecs.harvard.edu.

†International Computer Science Institute, Berkeley, CA 94704. e-mail: voecking@icsi.berkeley.edu. Supported by a grant of the "Gemeinsames Hochschulsonderprogramm III von Bund und Ländern" through the DAAD.

To motivate this paper, we first provide the relevant history. It is well known that when n balls are thrown into n bins, the maximum load, or balls in a bin, is $\frac{\ln n}{\ln \ln n}(1 + o(1))$ with high probability. Azar, Broder, Karlin, and Upfal suggested the following variation [1]. Suppose that n balls are sequentially placed into n bins in the following manner: for each ball, $d \geq 2$ bins are chosen independently and uniformly at random from the n bins, and the ball is placed in the bin with the fewest balls, ties being broken arbitrarily. Then in this case the maximum load is only $\frac{\ln \ln n}{\ln d} \pm \Theta(1)$ with high probability. This implies that two choices yields an "exponential improvement" over one choice, but three choices is just a small factor better than two. If we model a hash function by a perfectly random placement function, then this result says that if we hash items using two hash functions and place the item in the least loaded bucket, then we can dramatically reduce the maximum load (and hence the maximum search time) for an item. (Of course the average search time may increase, since a search requires examining multiple buckets.) We note that the paper [1] also examined several related problems, including a closed dynamic model where at each step a random ball is deleted and re-inserted into the system.

This result was generalized to natural queueing theoretic models independently by Vvedenskaya, Dobrushin, and Karpelevich [14] and Mitzenmacher [9, 10]. (See also the work by Eager, Lazowska, and Zahorjan [2].) The standard model is as follows. Suppose that tasks arrive at a bank of n First In First Out processors as a Poisson process of rate λn, where $\lambda < 1$. (Note the arrival rate per processor is a fixed constant.) Tasks require an exponentially distributed amount of service with mean 1. If each task queues at a random processor, then in the stationary distribution the probability that a server has at least k tasks is simply λ^k. If instead each task chooses two processors at random and queues at the shorter, than in the fluid limit process representing the limiting behavior as n grows, the probability that a server has at least k tasks converges to $\lambda^{\frac{d^k-1}{d-1}}$. That is, the tails of the processor queue lengths decrease *doubly exponentially*, rather than exponentially, when $d > 1$. This naturally leads to exponential gains in the average time in the system; moreover, the effect is clear in simulations even for relatively small values of n.

In this paper, we consider variations on these schemes, based on the work by Vöcking [13]. He considered the following variation of the original balls-and-bins problem: split the n bins into d groups of size n/d. For convenience, let us think of these groups as being laid out linearly from 1 to n, with the first group (bins 1 to n/d) being thought of being as the *leftmost*, and so on. A ball is placed as follows: a bin is chosen independently and uniformly at random from each of the d groups. A ball is placed into the least loaded of

d bins, but ties are broken by placing the ball in the leftmost bin. Vöcking showed that the maximum number of balls in a bin in this case is $\frac{\ln \ln n}{d \ln \phi_d} \pm \Theta(1)$ with high probability, where here ϕ_d corresponds to the exponent of growth for a generalized Fibonacci sequence. (We explain this later; for reference, $\phi_2 = 1.61 < \phi_3 < \phi_4 \ldots < 2$.) Surprisingly, then, coordinating ties in this manner improves performance! This model is also quite natural for hashing: when using multiple hash functions, the hash table is split into disjoint blocks, and each hash function provides a bucket from a specific block. These results also hold for dynamic models where the number of balls in the system is bounded, and deletions are controlled by an adversary of limited power; see [13] for more details. To differentiate the original multiple-choice scheme from the one recently presented in [13], we call the prior the *d-random* scheme and the latter the *d-left* scheme.

Here, following the work of [12], we examine the d-left scheme using a fluid limit model, which corresponds to a family of differential equations. We also examine the natural queueing generalization, where balls are tasks and bins are First In First Out queues. Besides providing natural intuition for the results in [13], the fluid limit model provides the correct limiting distributions for both the balls and bins scenario and the queueing scenario. Also the fluid limit models provide insights that lead to further small improvements in the d-left approach.

2 Balls and bins

We first review the fluid limit model for the d-random scheme by considering the case $d = 2$. Let $w_i(t)$ be the fraction of the n bins that have load at least i when tn balls have been thrown. Note $w_0(t) = 1$ always. We will drop the reference to t where the meaning is clear, using w_i in place of $w_i(t)$. For $i \geq 1$, the fluid limit describing the behavior of the w_i are

$$\frac{dw_i}{dt} = w_{i-1}^2 - w_i^2. \tag{1}$$

The intuition is that w_i increases when both of the bins chosen have load at least $i - 1$ but both do not have load at least i. That these differential equations accurately describe the behavior of the process follows from the framework established by Kurtz [3, 4, 5]. Here we will just assume that the intuitive differential equations are the proper fluid limit; further details and similar results can be found in for example [9, 12, 14].

For the static load balancing problem, we are most interested in the case of n balls and n bins. For the fluid limit model, this corresponds to the time

$t = 1$. As the w_i are increasing, we can bound their behavior using simple manipulation:

$$\begin{aligned}
\frac{dw_i}{dt} &\leq (w_{i-1}(t))^2; \\
w_i(1) &\leq \int_0^1 (w_{i-1}(t))^2 dt \\
&\leq \int_0^1 (w_{i-1}(1))^2 dt \\
&\leq (w_{i-1}(1))^2
\end{aligned}$$

From this, a simple induction yields that the w_i decrease *doubly exponentially*; that is, $w_i(1) \leq (w_1(1))^{2^{i-1}}$. This doubly exponential decrease of the tails is noteworthy, as it is an entirely different behavior than when each ball has a single random choice.

For general d, the fluid limit model is given by

$$\frac{dw_i}{dt} = w_{i-1}^d - w_i^d, \tag{2}$$

and we find that $w_i(1) \leq (w_1(1))^{d^{i-1}}$.

We now consider a fluid limit model for the d-left scheme, again starting with $d = 2$. Let $y_i(t)$ be the fraction of the n bins that have load at least i and are in the first, leftmost group when nt balls have been thrown. Similarly, let $z_i(t)$ be the fraction of the n bins that have load at least i and are in the second group on the right when nt balls have been thrown. Note $y_i(t), z_i(t) \leq 1/2$ and $y_0(t) = z_0(t) = 1/2$ for all time. If we choose a random bin on the left, the probability that it has load at least i is $\frac{y_i}{1/2} = 2y_i$. Analogously, if we choose a random bin on the right, the probability that it has load at least i is $2z_i$.

The fluid limit as n grows to infinity is given by the following differential equations, where again $i \geq 1$:

$$\begin{aligned}
\frac{dy_i}{dt} &= 2\left(y_{i-1} - y_i\right)\left(2z_{i-1}\right) ; \\
\frac{dz_i}{dt} &= 2\left(z_{i-1} - z_i\right)\left(2y_i\right).
\end{aligned}$$

That is, for y_i to increase, our choice on the left must have load $i - 1$, and the choice on the right must have load at least $i - 1$. For z_i to increase, our choice on the right must have load $i - 1$, but now the choice on the left must have load at least i.

It will be somewhat more convenient to generalize to the case of general d if we write these equations all in terms of a single sequence x_i. If we substitute

x_{2i} for y_i and x_{2i+1} for z_i, the equations above nicely simplify to the following (for $i \geq 2$):

$$\begin{aligned} \frac{dx_i}{dt} &= 2\left(x_{i-2} - x_i\right)\left(2x_{i-1}\right) \\ &= 4\left(x_{i-2} - x_i\right)x_{i-1}. \end{aligned} \tag{3}$$

Our results follow from this family of differential equations. We first demonstrate, in an admittedly somewhat non-rigorous fashion, how results similar to those of [13] are easily derived from the previous equations, using the fluid limits. Again, we run the system until time $t = 1$. Then

$$\begin{aligned} x_i(1) &= \int_{t=0}^{1} \frac{dx_i}{dt} dt \\ &= \int_{t=0}^{1} 4\left(x_{i-2}(t) - x_i(t)\right)x_{i-1}(t)dt \\ &\leq \int_{t=0}^{1} 4x_{i-2}(t)x_{i-1}(t)dt \\ &\leq \int_{t=0}^{1} 4x_{i-2}(1)x_{i-1}(1)dt \\ &\leq 4x_{i-1}(1)x_{i-2}(1) \end{aligned}$$

From this recursion we can derive closed form upper bounds in terms of the Fibonacci numbers. Let F_j represent the Fibonacci sequence, with $F(0) = 0$, $F(1) = 1$, and $F(k) = F(k-1) + F(k-2)$, for $k \geq 2$. Then a simple induction yields that for $i \geq 8$,

$$x_i(1) \leq 4^{F_{i-5}-1}(x_7(1))^{F_{i-6}}(x_6(1))^{F_{i-7}}.$$

At time $t = 1$, the number of bins with load at least 3 is at most $n/3$. Hence, $x_7(1) + x_6(1) \leq \frac{1}{3}$. As $x_7(1) \leq x_6(1)$, this implies that for $i \geq 8$

$$x_i(1) \leq 4^{F_{i-5}-1}6^{-(F_{i-6}+F_{i-7})} \ = \ \frac{1}{4} \cdot \left(\frac{2}{3}\right)^{F_{i-5}}.$$

(Note that this bound can be tightened, in that the fast decrease in the tails occurs before $i = 8$; the above is just a simple way to demonstrate how the tails behave.)

For $d \geq 2$, an entirely similar argument applies, except that one must use generalized Fibonacci numbers. That is, we may define the generalized Fibonacci number $F_d(k)$ by $F_d(k) = 0$ for $k \leq 0$ and $F_d(1) = 1$. When $k > 1$, $F_d(k) = \sum_{i=1}^{d} F_d(k-i)$. For the d-left scheme, we may think of x_{jd+k} as representing the fraction of the bins that have load at least j in the kth group from the left (where the leftmost group is the 0th group from the left). Then the fluid limit model yields the following family of differential equations:

$$\frac{dx_i}{dt} = d^d \left(x_{i-d} - x_i\right) \prod_{j=i-d+1}^{i-1} x_j. \tag{4}$$

By integrating the above from $t = 0$ to $t = 1$, we find

$$x_i(1) \leq d^d \prod_{j=i-d}^{i-1} x_j(1).$$

From this recursion, we can derive an upper bound in a closed form as follows. At time $t = 1$, the number of bins with load at least 3 is at most $n/3$. Therefore, $\sum_{i=3d}^{4d-1} x_i(1) \leq \frac{1}{3}$. Under this restriction, and the restriction that $x_i(1) \geq 0$ for all i, the values for the $x_k's$ with $k \geq 4d$ would be maximized by setting $x_{3d}(1) = \cdots = x_{4d-1}(1) = \frac{1}{3d}$. (These are of course not the actual values, but they suffice for an upper bound.) Hence, for $4d \leq i \leq 5d-1$, we have that

$$x_i(1) \leq d^d \left(\frac{1}{3d}\right)^d \leq \left(\frac{1}{3}\right)^d.$$

Then a simple induction yields that for $i \geq 4d$,

$$x_i(1) \leq (3d)^{-F_d(i-5d)+1}$$

Let $w_i'(1)$ denote the fraction of bins with load at least i in the d-left system. Then

$$w_i'(1) = \sum_{j=0}^{d} x_{di+j} \ \leq d(3d)^{-F_d((i-5)d)+1}$$

Hence for the tails in the d-left, the exponent of the fraction of bins with load at least i grows like $F_d(d \cdot i)$. We say that the tails in this instance fall *Fibonacci exponentially*, as opposed to exponentially or doubly exponentially.

To clarify this behavior, we define $\phi_d = \lim_{k \to \infty} \sqrt[k]{F_d(k)}$. For example, the value ϕ_2 corresponds to the golden ratio, $\phi_2 = 1.61....$ The ϕ_d are an increasing sequences, satisfying $2^{(d-1)/d} < \phi_d < 2$. Hence, that the tails fall Fibonacci exponentially for the d-left scheme implies that the tails $w_i'(1)$ satisfy

$$w_i'(1) \leq c^{\phi_d{}^{id}}$$

for a properly chosen $c < 1$ and sufficently large i. Note that this means the tails fall in a much faster manner for the d-left scheme than for the d-random

i	$d=2$		$d=3$		$d=4$	
	$w_i(1)$	$w'_i(1)$	$w_i(1)$	$w'_i(1)$	$w_i(1)$	$w'_i(1)$
1	7.6e-1	7.7e-1	8.2e-1	8.4e-1	8.6e-1	8.8e-1
2	2.3e-1	2.2e-1	1.8e-1	1.6e-1	1.4e-1	1.2e-1
3	8.9e-3	4.4e-3	5.1e-4	1.1e-5	2.3e-5	7.8e-11
4	6.0e-6	5.2e-8	3.9e-12	4.5e-31	4.0e-21	2.3e-141

Table 1: Results for d-left (w') vs. d-random (w) in the fluid limit model.

scheme. (It is worth noting that one can derive similar lower bounds for the d-left and d-random schemes, so their tail behaviors are really Fibonacci exponential and doubly exponential, respectively.)

In order to get very exact numerical estimations on the fraction of bins with a given number of balls after n balls have been thrown into n bins, it suffices to simulate equation 3 over 1 unit of time (using sufficient arithmetic precision). We compare results for the d-left and d-random systems in Table 1, which was generated from a C program (and checked via a separate Maple program). Obviously, the d-left scheme has more rapidly decreasing tails, implying that the maximum load tends to be less in the d-left scheme. From the results in Table 1 we can conclude that using the 2-random scheme, approximately six out of one million balls would have height at least four, whereas in the d-left scheme, only five out of one hundred million balls would have height at least four. Hence one would expect that when one million balls are thrown into on million bins, the 2-random scheme would have some balls of height four, whereas using the 2-left scheme this would happen much less often. Simulations bear out these results. We simulated the process of throwing one million balls into one millions bin three hundred times (three separate runs of one hundred simulations). In every case, using the 2-random scheme, the maximum load was four. Using the 2-left scheme, the maximum load was four in only twenty of the three hundred runs.

In correspondence to our calculations above (and to the results in [13]) this effect becomes more obvious when d is increased. For example, one can expect that the 4-random scheme produces a maximum load of three when the number of balls and bins is in the hundreds of thousands, whereas the 4-left scheme does not place more than 2 balls in the same bin with very high probability unless the number of bins is in the billions!

An interesting variation of the original scheme arises naturally from considering the differential equations. Consider again the specific case $d = 2$. There is no reason that we necessarily have to break the left and right sides

i	$w_i(1)$	$w'_i(1)$ $\alpha = 0.5$	$w'_i(1)$ $\alpha = 0.585$
1	0.7616	0.7717	0.7758
2	0.2295	0.2239	0.2201
3	0.0089	0.0045	0.0041
4	6.0e-6	5.2e-8	3.8e-8
5	1.3e-12	1.2e-21	6.1e-22
6	3.2e-26	5.3e-58	8.8e-59

Table 2: Results for 2-left (w') vs. 2-random (w) in the fluid limit model.

into equal parts. Although this variation was not considered in [13], it does not affect the asymptotic bounds presented there. With the fluid limit, we show this variation can improve performance. Intuitively, the improvement occurs because breaking ties toward the left also increases the average number of balls per bin on the left, raising the load on these bins; by skewing the division, we can keep the structure we obtain from breaking ties regularly while maintaining a more even load across all bins.

Suppose we split the bins so that the left contains $\alpha \cdot n$ bins, and the right contains $(1-\alpha) \cdot n$ bins. Then $x_0 = \alpha$, $x_1 = 1-\alpha$ for all time, and by the same reasoning as for equation (3),

$$\frac{dx_i}{dt} = \frac{1}{\alpha(1-\alpha)} \left(x_{i-2} - x_i\right) x_{i-1} \tag{5}$$

By simulating the differential equations to $t = 1$, we find that the tails decrease faster as α increases up to about 0.585. The results for $\alpha = 0.585$ are presented for comparison in Table 2.

Experiments further suggest that as the number of balls placed in the system increases (so that t grows larger than 1), the value of α that minimizes the maximum load decreases to 1/2. Currently we do not know of any means for comparing systems with the same number of choices but differing values of α other than by simulating the underlying differential equations, or by simulating the actual system for a specific value of n. We can determine the proper choice of α only by experiment; finding better approaches remains an open question.

For general d, if the ith group from the left consists of a fraction α_i of the bins, then the equations become

$$\frac{dx_i}{dt} = \left(\prod_{k=0}^{d-1} \frac{1}{\alpha_k}\right) \left(x_{i-d} - x_i\right) \prod_{j=i-d+1}^{i-1} x_j. \tag{6}$$

3 Dynamic Models

We briefly review the standard dynamic model; for further details, see [14] or [9, 10]. In the standard dynamic model, we think of the bins as First In First Out servers. We think of the balls as tasks that arrive as a Poisson process with interarrival time $\frac{1}{\lambda n}$, for some $\lambda < 1$. Each task has an exponentially distributed service time with mean 1. Suppose arriving tasks choose a processor according to the d-random scheme. Again, we let w_i represent the fraction of the bins with at least i tasks in the queue (including the one being served). Then the differential equations describing the fluid limit are

$$\frac{dw_i}{dt} = \lambda(w_{i-1}^d - w_i^d) - (w_i - w_{i+1}). \tag{7}$$

Of course $w_0(t) = 1$ always. This equation arises from the following natural intution: arrivals occur at a rate λ per server, and an arrival is the ith task at a queue when all of its choices have at least $i-1$ tasks but not all have at least i tasks. This occurs with probability $w_{i-1}^d - w_i^d$. Departures from servers with i tasks occur at rate 1, and the fraction of servers with i tasks is $w_i - w_{i+1}$.

The key to understanding these systems generally revolves around their *fixed points*, where all the derivatives are zero. Moreover, as we expect the system to be stable, we seek fixed points where the w_i fall to 0 as i grows and the expected number of tasks in the system remains finite. It has been shown that the dynamic d-random system has a unique fixed point satisfying this condition. Let us denote it by w_i^*; then the fixed point is given by $w_i^* = \lambda^{\frac{d^i - 1}{d-1}}$. It has been shown that the the trajectories of the limiting system converge to the fixed point over time [10, 14], and the stationary distribution for systems with large n are concentrated around this fixed point [14]. Hence, we find that even for $d = 2$, the tails of the queue lengths decrease doubly exponentially. If instead each task chooses a random server, then each server behaves like an M/M/1 queue, and the tails of the queue lengths decrease exponentially as λ^i. Just two choices therefore yields dramatically better performance [9].

We now examine the improvements obtained using the d-left scheme. A natural hypothesis is that there will be some gain over the d-random scheme, but that it will be small, since the only difference is in case of ties. Again, we first consider the case $d = 2$. We obtain the fluid limit equations

$$\frac{dy_i}{dt} = 4\lambda\left(y_{i-1} - y_i\right) z_{i-1} - \left(y_i - y_{i+1}\right)\ ; \tag{8}$$

$$\frac{dz_i}{dt} = 4\lambda\left(z_{i-1} - z_i\right) y_i - \left(z_i - z_{i+1}\right), \tag{9}$$

where y_i and z_i represent the fraction of servers with queue length at least i on the left and right, respectively. As before, we may simplify for general d by using x_{jd+k} to represent the fraction of servers that have queue length at least j and lie in the kth group from the left. Then the fluid limit model yields the following family of differential equations:

$$\frac{dx_i}{dt} = d^d \lambda \left(x_{i-d} - x_i\right) \left(\prod_{j=i-d+1}^{i-1} x_j \right) - (x_i - x_{i+d}). \tag{10}$$

This system appears substantially more complex than the standard model, even in the simplest case where $d = 2$. Indeed, we have not been able to prove that there is a unique fixed point (satisfying $x_i \to 0$ as $i \to \infty$), although that appears to be the case based on simulating the process determined by the differential equations. We will from here on assume that there is indeed such a unique fixed point for this system. (Note, however, that for any fixed number of servers the process is clearly stable, by a simple stochastic comparison with a system where each task simply chooses a random server. In this case, each server behaves like a standard M/M/1 server.)

With this assumption, we can prove that the tails for this system at the fixed point are term by term no greater than the tails at the fixed point of the d-random system. Let $u_i = \sum_{k=0}^{d-1} x_{id+k}$. That is, u_i is the total fraction of servers with load at least i. Clearly $u_0 = 1$. We represent the values at the fixed point for the d-left scheme by u_i^* and x_i^*. Since all derivatives are zero at the fixed point, we have

$$d^d \lambda \left(x_{i-d}^* - x_i^*\right) \left(\prod_{j=i-d+1}^{i-1} x_j^* \right) - (x_i^* - x_{i+d}^*) = 0. \tag{11}$$

Summing these expressions from $i = jd$ to infinity yields

$$u_j^* = d^d \lambda \prod_{i=(j-1)d}^{jd-1} x_i^*.$$

Hence, for instance, it follows at the fixed point that $u_1^* = \lambda$, which corresponds to the obvious condition that the arrival rate must equal the departure rate at the fixed point. But now by the arithmetic-geometric mean inequality

$$u_j^* = d^d \lambda \prod_{i=(j-1)d}^{jd-1} x_i^* \leq \lambda \left(\sum_{i=(j-1)d}^{jd-1} x_i^* \right)^d = \lambda \left(u_{j-1}^*\right)^d.$$

A simple direct induction now yields that $u_j^* \leq \lambda^{\frac{d^j-1}{d-1}}$, and hence in equilibrium in the fluid limit the d-random system has larger tails than a d-left system.

As in the static case, we find that the nature of the improvement comes from the different behavior of the tails. At the fixed point, by summing dx_i/dt from $i = k$ to infinity yields

$$\sum_{j=k}^{k+d-1} x_j^* \leq d^d \lambda \prod_{h=k-d}^{k-1} x_h^* \tag{12}$$

and hence

$$x_k^* \leq d^d \lambda \prod_{h=k-d}^{k-1} x_h^*. \tag{13}$$

By the same argument as we used for the balls and bins problem, this condition is sufficient to show that at the fixed point for the dynamic d-left scenario, the tails decrease Fibonacci exponentially according to the generalized Fibonacci numbers.

We can again improve the situation by changing the ratio of left and right bins. When $d = 2$ the equations become

$$\frac{dx_i}{dt} = \frac{1}{\alpha(1-\alpha)} \lambda \left(x_{i-2} - x_i\right) x_{i-1} - (x_i - x_{i+2}), \tag{14}$$

with the conditions $x_0 = \alpha$ and $x_1 = 1 - \alpha$. Naturally, in this situation, the best α is a function of the arrival rate per server λ.

We present results obtained from finding the fixed point from the fluid limit model through simulating the differential equations. In Table 3, we show the expected time a customer spends in the system, a quantity easily derived from the calculation of the fixed point. We focus first on the expected time in the system as it is a natural measure of overall performance. Comparing d-left and d-random systems for the case $d = 2$, we find that d-left systems offer slightly better performance. We also searched for the best α value, finding it to the nearest thousandth. As can be seen, choosing the right value of α also offers a marginal improvement, yielding gains of approximately 1% over d-random systems. Although this gain is small, it should be emphasized that the d-random scheme already performs vastly better than simple random selection of queues; hence even this small improvement is interesting.

It is worthwhile to ask whether the gains apparent in the limiting analysis manifests in systems of reasonable size. We explore a specific data point in detail. Consider a system with 100 servers, and a Poisson arrival process with an average of 90 arrivals per unit time. That is, $\lambda = 0.9$. In simulations, we ran such a system for 50,000 units of time, recording the time spend in the system for every task. Results from the first 5,000 units of time are discarded in order to allow the system to avoid bias from starting at an empty state. The

λ	2-random	2-left $\alpha = 0.5$	2-left best	α
0.5	1.26569	1.25325	1.24880	0.572
0.6	1.40744	1.39433	1.38934	0.561
0.7	1.61445	1.60054	1.59505	0.550
0.8	1.94736	1.93234	1.92627	0.540
0.9	2.61406	2.59727	2.59028	0.529

Table 3: Expected time in the system from the fluid limit model for 2-left vs. 2-random.

	2-random	2-left $\alpha = 0.5$	2-left $\alpha = 0.53$
Average time	2.64626 ± 0.00148	2.63251 ± 0.00132	2.62485 ± 0.00160
Sample std. dev.	0.00756	0.00673	0.00817

Table 4: Improvements by using 2-left and an appropriate α, at $\lambda = 0.9$ and 100 servers.

results based on simulating each system 100 times are presented in Table 4. The average time in the system is given, along with 95% confidence intervals, from our simulations. Although the differences are, as expected, small (less than 1%), they clearly appear statistically significant. For smaller arrival rates (down to $\lambda = 0.5$), the difference between the best 2-left scheme and the 2-random scheme is approximately 1%, as suggested by the results in Table 3.

The improvement from using the d-left scheme becomes more pronounced if we consider a different measure of performance other than the expected time in the system. As we have shown, the d-left scheme leads to a faster decrease in the tails of the loads. Hence, the main benefit of the d-left over the d-random scheme is that the probability of joining a long queue, and thereby spending a long time in the system, is much smaller under the d-left scheme. Indeed, this is one would naturally expect from the static case. We demonstrate this behavior in Table 5 by focusing on the case $\lambda = 0.9$ and comparing fixed point values for the d-left and d-random schemes. (Values under 1.0e-100 are left blank.)

4 Related models

It is worthwhile to consider related models that have used the d-random strategy and consider the effect of instead using the d-left strategy. For example,

i	$d=2$		$d=3$		$d=4$	
	w_i^*	u_i^*	w_i^*	u_i^*	w_i^*	u_i^*
1	9.0e-1	9.0e-1	9.0e-1	9.0e-1	9.0e-1	9.0e-1
2	7.3e-1	7.3e-1	6.5e-1	6.5e-1	5.9e-1	5.9e-1
3	4.8e-1	4.8e-1	2.5e-1	2.5e-1	1.1e-1	9.8e-2
4	2.1e-1	2.0e-1	1.4e-2	9.1e-3	1.3e-4	5.3e-7
5	3.8e-2	3.2e-2	2.9e-6	5.8e-10	2.5e-16	3.4e-77
6	1.3e-3	5.8e-4	2.9e-17	1.4e-54	3.5e-63	
7	1.5e-6	2.5e-8	9.7e-51			

Table 5: Results for d-left (u^*) vs. d-random (w^*) in the fluid limit model when $\lambda = 0.9$.

recent work by Martin and Suhov [6] has extended the d-random approach to Jackson-like networks consisting of banks of servers. The nodes of the Jackson networks are now banks of queues, and when an incoming task enters a bank of queues, it adopts the d-random strategy. They call such a network a Fast Jackson Network. (See also [7, 8].) Just as a Jackson network has a simple product-form stationary distribution, so too does the limiting model of the fast Jackson network, where the size of the server banks grows to infinity. Each server banks has a stationary distribution where the tails decrease doubly exponentially.

If one could prove that the fluid limit of a bank of servers using the d-left strategy converges appropriately, it is likely that the analysis of Martin and Suhov would immediately extend to Jackson-like networks of banks of servers where incoming tasks use the d-left strategy. For such a network, the tails at each server would decrease more rapidly (Fibonacci exponentially), and hence the d-left strategy would yield a (slightly) Faster Jackson Network.

The d-random strategy has also been studied in cases where the load information one obtains is out of date. For example, consider a bank of servers where the queue length information is updated only at intervals of some time T. In this setting, choosing the shortest of two random servers can perform even better than the strategy of choosing a server with the apparently shortest queue; when all incoming tasks try to choose the shortest queue, the delay leads to repeated situations where only a few servers take tasks until an update occurs. We argue that in this situation, the d-left strategy is similarly apt to perform poorly, especially under long delays T. The problem is that breaking ties in the same way again pushes the load toward a smaller set of servers. Because the updates are delayed, this effect is not noticed immediately by new tasks, leading to higher loads. Experiments verify this intuition. Note, however, that generally high arrival rates and long delays are required before a substantial effect is seen, since the difference between the two strategies is otherwise small.

References

[1] Y. Azar, A. Broder, A. Karlin, and E. Upfal. Balanced Allocations. In *Proceedings of the 26th ACM Symposium on the Theory of Computing*, 1994, pp. 593–602.

[2] D. L. Eager, E. D. Lazowska, and J. Zahorjan. Adaptive load sharing in homogeneous distributed systems. *IEEE Transactions on Software Engineering*, Vol. 12, 1986, pp. 662–675.

[3] S. N. Ethier and T. G. Kurtz. **Markov Processes: Characterization and Convergence**. 1986, John Wiley and Sons.

[4] T. G. Kurtz. Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes. *Journal of Applied Probability* Vol. 7, 1970, pp. 49-58.

[5] T. G. Kurtz. **Approximation of Population Processes**. SIAM, 1981.

[6] J. B. Martin and Y. M Suhov. Fast Jackson Networks. To appear in *Annals of Applied Probability.*

[7] J. B. Martin. Stochastic Bounds for Fast Jackson Networks. Submitted for publication.

[8] J. B. Martin. Point Processes in Fast Jackson Networks. Submitted for publication.

[9] M. Mitzenmacher, The Power of Two Choices in Randomized Load Balancing. Ph.D. thesis, University of California, Berkeley, September 1996.

[10] M. Mitzenmacher, Load Balancing and Density Dependent Jump Markov Processes. In *Proc. of the* 37^{th} *IEEE Symp. on Foundations of Computer Science*, 1996, pp. 213–222.

[11] M. Mitzenmacher, How Useful is Old Information? In *Proc. of the 16th ACM Symp. on Principles of Distributed Computing*, 1997, pp. 83–91.

[12] M. Mitzenmacher, Studying Balanced Allocations with Differential Equations. To appear in *Combinatorics, Probability, and Computing.*

[13] B. Vöcking. How Asymmetry Helps Load Balancing. To appear in FOCS '99.

[14] N.D. Vvedenskaya, R.L. Dobrushin, and F.I. Karpelevich. Queueing System with Selection of the Shortest of Two Queues: an Asymptotic Approach. In *Problems of Information Transmission*, Vol 32, 1996, pp. 15–27.