Performance of Shared Memory in a Parallel Computer

Kevin Donovan

March 1994

Abstract

Suppose that in a single memory cycle, n independent random accesses are made to m separate memory modules, with each access equally likely to go to any of the memories. Let L_{avg} then represent the expected value of the maximum number of references to a single memory module. Here we show a new method for analyzing this problem. It allows one to efficiently compute narrow upper and lower bounds for L_{avg} as a function of m and n. We also determine the asymptotic behavior of L_{avg} as m and n grow to infinity at a constant ratio $\lambda = n/m$. For any $\lambda > 0$, this paper proves that $L_{\text{avg}} = (1 + o(1)) \log m / \log\log m$ as m and $n \to \infty$. An equivalent result was previously obtained by Gonnet in connection with a hashing problem. Using different methods, Gonnet found the asymptotic value of $\Gamma^{-1}(m)$ (plus lower order terms).

Index Terms—Asymptotic behavior, combinatorial analysis, crossbar networks, MIMD computers, performance evaluation.

1 Introduction

The particular application that motivated this study is the performance analysis of parallel computers, especially vector machines in which processors and memories are connected by a crossbar. This means there is a communication path between each processor and memory that does not conflict with the path between any other processor and memory. However, if a memory module is addressed by more than one processor during an instruction cycle, the different accesses must be serviced sequentially, and the program cannot advance until all memory requests are satisfied. In such a case, the time to perform an instruction increases linearly with the length of the maximum request queue. Consequently, the hardware designer wishes the memory requests to be spread as uniformly as possible on average.

In typical cases, the programmer controls how the data are distributed across memory modules. Ideally, the programmer tries to structure the data and references so that no one memory contains more than one of the items accessed in a single instruction cycle. E.g., if two vectors are added, and the addition proceeds sequentially through the vector, then the first item of the vector can be kept in the first memory module, the second in the second, and so forth. Some vector operations do not have such a simple sequence of accesses, however. In order to allow efficient access for a variety of such sequences, the programmer may use a hashing scheme to spread the data across memories. The data may be effectively randomly distributed across memory modules, and we wish to know the expected length of the maximum memory queue in such a case.

As hardware costs go down, the number of memory modules in a system can go up. Naturally, the behavior of maximum queue length in a system with four or eight memories will be different from that of one with 64 or 256 or more. One might wish to know how the ratio of processors to memories affects the queue length. In this paper, we look at what happens to this value for large numbers of processors and memories.

Give n processors and m memories, the problem is analyzed here by looking at the probability distribution of the maximum memory queue length. We derive a recurrence relation for the probabilities, and find inequalities satisfied by the recurrence. These inequalities allow us to efficiently determine narrow lower and upper bounds for the probability distribution function. Moreover, the upper and lower bounds approach a common limit as n and m grow to infinity at a constant ratio $\lambda = n/m$; By finding an expression for this limit, we determine the asymptotic behavior of L_{avg}, the maximum queue length.

We obtain the following result. Given any positive rational number λ, let there be m memories and λm processors, and let m grow to infinity while λ stays fixed. We find that the expected value L_{avg} grows to infinity with m. More precisely, for any $\lambda > 0$, we show that

$$L_{\text{avg}} = \frac{\log m}{\log\log m}\bigl(1 + o(1)\bigr) \qquad \text{as } m \to \infty.$$

The correction factor is not necessarily close to unity for practical values of m and n. So we show graphs of L_{avg} for m in the range 100–100,000, and $0.25 \le \lambda \le 4.0$. The graphs confirm that the growth rate of L_{avg} is approximately the same as that indicated by the formula above, and the graphs accurately describe the magnitude of L_{avg} for this range of m and n. We also prove that the probability distribution of the maximum queue length becomes increasingly concentrated as m and n grow to infinity at a constant ratio. When m and n are large, the maximum queue length is very likely to be equal to either $\lfloor L_{\text{avg}} \rfloor$ or $\lceil L_{\text{avg}} \rceil$. We also report on simulations made to verify the analysis that led to these results.

This problem falls in the general category of urn problems. A good survey of related problems can be found in Johnson–Kotz [1977]. The asymptotic behavior of a similar problem was studied in Klamkin–Newman [1967] and extended by Dwass [1969]. In the terminology used here, they looked at what happened when the number of memories was held fixed, and considered how many references would need to be made in order for the expected value of the maximum queue length to be r. This was not the growth pattern we were interested in (we wanted to know what happens when m and n grow together), and these papers did not investigate the error terms nor the speed of convergence.

Flajolet [1983] sharpened and extended the estimates made by these authors. The application addressed by Flajolet relates to trie searching, particularly the maximum depth of the trie directories. Like the previous papers, Flajolet [1983] obtains the number of memories as a function of the number of processors and the maximum queue length, whereas we obtain the queue length as a function of the number of processors and memories. Otherwise, the two problems are the same.

However, Flajolet [1983] uses methods entirely different from those we will be presenting. Here, we set up a recurrence relation and find inequalities based on the recurrence. In contrast, Flajolet [1983] makes use of the fact that the generating function is an analytic function, in fact a polynomial. He constructs an integral whose value is the solution being sought, and he estimates the value of the integral. He proves that the error terms of the estimates asymptotically go to zero, and he presents graphical results showing that one form of the approximate solution gives close results even when m and n are small.

We attempted to write a computer program based on the analysis shown in Flajolet [1983], and to compare his results with ours. We ran into some difficulties in writing the program, however. We were able to get this other approach to work only for certain values of m and n, including the cases $m = n \geq 2048$, and also when $m = 10^7$ and $n = 10^9$. In all cases, the results found by the method that will be derived here agreed closely with those found using our implementation of Flajolet's approach. Our programs for the two different approaches—Flajolet's and that presented here—were about equally efficient.

Finally, an equivalent problem was investigated by Gonnet [1983] with respect to hashing with separate chaining. Gonnet considered how long hash chains grow if n hash items are distributed among m hash buckets, with hash conflicts handled by chaining. By identifying processors with hash items, memories with hash buckets, and memory queues with hash chains, the problems are immediately seen to be the same.

Several new results are presented in this paper. This appears to be the first work that looks at the problem from the point of view of finding the maximum queue length in terms of the numbers of processors and memories, and consequently, the limit theorem we present is new. The method of analysis we present here is a new approach to this problem. Both our approach and Flajolet's are based on estimates of the error terms of approximations, and the two derivations appear equally complex. Gonnet used a function-theoretic approach which appears to be roughly as complex as the other two methods; a disadvantage of his approach is that it looks only at the average value, rather than the probability distribution function, although it may be that Gonnet simply chose to study the more restricted problem. Previous authors, including Flajolet and Gonnet, commented on the sharpness of the probability distribution function, but this appears to be the first paper that proves this property to hold asymptotically. Finally, our numeric results contain both upper and lower bounds, whereas previous authors had considered only the asymptotic behavior.

In the section following this one, we define the basic terminology, and then we present numerical and graphical results of this study. Next, in Sections 4–7, we derive the approximate method for determining the maximum queue length. This method gives upper and lower bounds, rather than exact answers. However, the bounds are close, and the running time for each (m, n)-case is only a few seconds. In the analysis, we first obtain a recurrence relation, in Section 4. Then, in Section 5, we show that each term of the recurrence satisfies certain inequalities. Section 6 discusses how these inequalities can be used in an algorithm to find upper and lower bounds of the expected maximum queue length. Afterwards, in Section 7, the recurrence and the inequalities are used to determine the asymptotic behavior of the probability distribution function, when n and m grow to infinity at a constant ratio $\lambda = n/m$. The final section of the paper contains a summary and conclusions.

2 Notation and Terminology

We first list some of the standard mathematical notation and conventions followed here. All sets considered in the paper are finite, and if A is a set, then $|A|$ represents the number of elements that A contains.

If r is a real number, then $\lfloor r \rfloor$ represents the floor function of r (greatest integer $\leq r$), and $\lceil r \rceil = -\lfloor -r \rfloor$ the ceiling function. The binomial coefficients $\binom{r}{k}$ are defined as in Knuth [1973]. The falling factorial function is represented by $[r]_k$, and is defined by $[r]_k = r(r-1)\cdots(r-k+1)$. For any real r, $\binom{r}{0} = [r]_0 = 1$, and $\binom{r}{k} = [r]_k = 0$ if $k < 0$.

Following Knuth [1973], the limits of a summation are not shown if these include all terms for which the defining expression is nonzero. The advantage of this practice is that one need not keep track of how the limits change when an expression is substituted for the index variable.

If f is an expression, then Df is the derivative with respect to the independent variable, which is normally t here. $D^n f$ is the nth derivative, and $D^0 f = f$. The symbol "log" refers to the natural logarithm: $\log x \equiv \log_e x$.

There now comes the terminology relevant to this particular application. First, the three variables that appear most frequently:

m – Number of memories (or letters in alphabet; see below).

n – Number of processors (or length of word).

r – Maximum references to any one memory (or maximum repetitions of any letter in a word).

Since any processor can access any memory independently, there are m^n different *access patterns* of processors to memories.

The standard terminology for a problem of this sort, as in Johnson–Kotz [1977], is to speak of randomly distributing n balls into m urns or boxes. In these terms, the quantity of interest, here called r, is the maximum number of balls in any single urn. Another equivalent representation that will sometimes be used in this paper, is that of words of length n, with letters drawn randomly from an alphabet containing m

letters. We are interested in the maximum number of occurrences of any of the m letters. The reason for using words rather than processors and memories is that it is easier to display a word than it is to draw a diagram. Consequently, much of the analysis in this paper will be phrased in terms of the word model, although we will always point out what the corresponding concept is in terms of processors and memories.

As an example, suppose $n = 3$ and $m = 2$. Let the processors be numbered 1, 2, and 3, and label the two memories a and b. The word baa corresponds to the access pattern where the first processor accesses memory b, and the other two processors access memory a. The total number of words is $m^n = 8$; they are $\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$. Of these, aaa and bbb each contain a letter that occurs a maximum of three times, so that the quantity r we are interested in is equal to 3. In the other six words, either a or b occurs a maximum of two times, and the quantity of interest is 2.

In the analysis that follows, we will be using a cumulative version of this quantity. Using the word representation, we will be looking at the number of words for which the maximum number of occurrences of any letter is less than or equal to r (rather than just equal to r). Because it is frequently used, we give a name to this quantity:

$Q(n, m, r)-$ Number of words containing n letters, drawn from an m-letter alphabet, for which no letter occurs more than r times.

In terms of processors and memories, $Q(n, m, r)$ is the number of ways that n processors can access m memories, with no more than r references to any one memory. We are assuming that all access patterns are equally likely. Consequently, if we divide by the total number of access patterns m^n, then the quotient $Q(n, m, r)/m^n$ is the probability that there are no more than r accesses to any one memory. In other words, if we can characterize $Q(n, m, r)$, then our problem is solved. This is in fact what we will be doing in the later sections of the paper.

Continuing with the above example where $n = 3$ and $m = 2$, we have $Q(3, 2, 1) = 0$, since no such word contains only a single a and a single b. $Q(3, 2, 2) = 6$, because there are six words for which a or b occurs twice, and $Q(3, 2, r) = 8$ if $r \geq 3$, because there a total of eight words, and in all eight of them, no letter occurs more than three times.

3 Data and Analysis

Before presenting the derivation of the calculations, we show numerical and graphical results.

In Figures 1–4 there appear graphs plotting the results for 104 different values of m, the number of memories. The variable m runs from 16 to 120194, with equal spacing on a logarithmic scale; each new value of m is $2^{1/8}$ times bigger than the previous value. Since m is an integer, we round each value to the nearest integer.

We made three different sets of runs for these values of m. Letting n be the number of processors and $\lambda = n/m$ the ratio of processors to memories, we made runs for $\lambda = 0.25$, 1.00, and 4.00. For the case $\lambda = 0.25$, since n must be an integer, we set $n = \lfloor m/4 + 0.5 \rfloor$. In each case, we computed upper and lower bounds to the expected maximum queue length, and to the probability distribution function.

The bounds for the expected maximum queue length is shown in Figure 1 on a semilogarithmic scale. For each value of λ, the graph is close to a straight line, which indicates that the maximum queue length increases at a logarithmic rate with increasing m for this range of values. In fact, we will be showing in Section 7 that the rate of growth is $o(\log m)$ as $m \to \infty$, which would correspond to a curve whose slope decreases with increasing m. For the two higher pairs of curves, the decrease in slope is visible for $\lambda = 1.0$ or $\lambda = 4.0$, but the effect is very slight for this range of numbers.

It is apparent from this graph that the lower and upper bounds become closer as m increases. For $m > 2000$, the curves almost overlap when drawn to the scale of Figure 1.

The next three figures show pairs of three-dimensional curves obtained from the same data. Each pair of curves shows the upper bound of the probability distribution function for a different value of λ. The first of these pairs, Figure 2, shows data for $\lambda = 0.25$. The left axis indicates the number of memories, on a logarithmic scale. The front axis is the same as the variable we have been calling r, drawn to linear scale. It represents the length of the maximum memory queue. The vertical axis is linear with probability.

For example, in the upper graph in Figure 2, the probability distribution function for $m = 16$ is shown at the back of the graph. From the graph, the case $r = 1$ is most likely; i.e., if $m = 16$ and $\lambda = 0.25$ (so that $n = 4$), then we can expect the length of the maximum queue to be one in the majority of cases. On the same graph, the case $m = 120194$ is shown at the front of the curve. Here, we see that the most likely cases are $r = 3$ or $r = 4$, with the latter being a little more likely.

The bottom curve of Figure 2 shows the same data as the top curve, but drawn from a different perspective. We have lowered our angle of sight, and we are looking directly at the two-dimensional cross section showing results for $m = 120194$. Figure 3 is just like this previous curve, but with results for $\lambda = 1.0$. Similarly, Figure 4 shows data for $\lambda = 4.0$. The important characteristic of all three graphs is the sharpness of the peak of the probability density function. The distribution is concentrated at one or two values of r if $\lambda \leq 1$, and at approximately five values of r when $\lambda = 4.0$.

In Table 1, we show a selection of more complete data. The seven headings indicate the following. "Procs" and "Mems" are the numbers of processors and memories, respectively. The remaining columns

Procs	Mems	Simul	FFT	Low	High	r_a
2	2	1.53	1.50	1.50	1.50	1.00
4	4	2.12	2.13	2.08	2.13	1.46
8	8	2.65	2.60	2.47	2.63	2.07
16	16	3.10	3.10	2.90	3.28	2.61
32	32	3.55	3.53	3.40	3.60	3.11
64	64	3.95	3.96	3.88	4.09	3.57
128	128	4.37	4.38	4.32	4.46	4.00
256	256	4.72	4.77	4.74	4.81	4.41
512	512	5.15	5.16	5.14	5.25	4.81
1024	1024	5.56	5.53	5.52	5.55	5.20
2048	2048	5.88	5.88	5.88	5.90	5.57
4096	4096	6.23	6.25	6.24	6.28	5.93
8192	8192			6.58	6.59	6.28
16384	16384			6.92	6.92	6.62
32768	32768			7.26	7.27	6.96
65536	65536			7.56	7.57	7.29
1024	512	7.48	7.48	7.46	7.58	6.71
1024	128	16.22		16.16	16.45	13.10
128	1024	2.27		2.26	2.27	2.49
10^9	10^9			11.63	11.63	11.41
10^7	10^9			3.34	3.34	3.37
10^9	10^7			157.34	157.47	148.45

Table 1. Maximum Expected Queue Length as Calculated in Different Ways

each show the expected value of the maximum queue length, when calculated in different ways. The column labeled "Simul" contains data obtained from a simulation. "FFT" is the value calculated using the fast Fourier transform to find the desired coefficient of the generating function. This method of solution does not use any approximations, and so the numbers in this column are exact except for rounding error. "Low" and "High" are lower and upper bound of the average queue length, calculated according to the algorithm that is derived in this paper. Finally, r_a is an easily calculated "asymptotic" approximation to the true value; it is defined in Section 7. A blank entry in a column indicates the corresponding item was not calculated for this set of n and m. Since the calculation of the FFT values is exact except for rounding error, it must be (and is) always within the range of values labeled Low and High.

We pass now to the derivation of the algorithm from which these numbers were calculated.

4 Recurrence Relation

The next several sections of the paper develop an approach to this problem that gives an efficient approximate solution, as well as the asymptotic behavior of the probability distribution function of the maximum queue length.

As a first step in the analysis, we find a generating function for $Q(n, m, r)$, which in this case is a polynomial that can be algebraically manipulated. Working with the generating function, we find a recurrence relation for $Q(n, m, r)$, or rather for a normalized variant of it, here called $P(n, m, r)$. The recurrence takes the form of a sum of terms, each of which contains two nonnegative factors. In the section that follows this one, we prove that these factors, and hence each term of the recurrence, satisfy certain inequalities. By using these inequalities in the recurrence relation, we are able to take one of the two factors of the summation outside. In most cases we are interested in, the factor that remains is very small except for a few terms. Consequently, we need consider only a few terms, and we can use the inequalities to estimate the terms that are omitted. We are able to obtain both upper and lower bounds on the true value. Section 6 shows how this is done in the computer algorithm.

Then, in Section 7 we take the inequalities and pass to the limit. There are various ways that n and/or m can grow to infinity. The case we study, and obtain a limit theorem for, is where n and m grow together at a constant ratio $\lambda = n/m$. We obtain the results that have already been described in the introduction to this paper.

Many books discuss generating functions, including Knuth [1973]. The generating we will now derive has also been used by previous authors working on this problem, including Flajolet [1983] among those who are cited here.

For the derivation of the generating function, let i_k represent the number of accesses to the kth memory. Since each of the n processors accesses a memory, the sum of the i_k must be equal to n. If none of the memories is accessed by more than r processors, then all of the i_k are less than or equal to r. Furthermore, if $i_k = l$, then the l processors accessing the kth memory can be chosen in $\binom{n}{l}$ different ways. If all of the i_k are specified, then the number of ways that they can be chosen is equal to the multinomial coefficient that appears in the formula below. The total number of such access patterns, $Q(n,m,r)$, is then found by summing over all combinations of the i_k that satisfy the constraints:

$$Q(n,m,r) = \sum_{\substack{i_1+i_2+\cdots+i_m=n \\ 0\le i_k\le r,\forall k}} \binom{n}{i_1\ i_2\ \cdots\ i_m} = \sum_{\substack{i_1+i_2+\cdots+i_m=n \\ 0\le i_k\le r,\forall k}} \frac{n!}{i_1!\, i_2! \cdots i_m!}.$$

By the multinomial theorem, the $Q(n,m,r)$ appear as the coefficients of the following polynomial:

$$\left(1+\frac{t}{1!}+\cdots+\frac{t^r}{r!}\right)^m = \sum_{n\ge 0} Q(n,m,r)\frac{t^n}{n!}.$$

If this expression is evaluated at $t = 0$, only the constant term $Q(0,m,r)$ remains. Evaluating the first derivative at $t = 0$ gives $Q(1,m,r)$, and in general, $Q(n,m,r)$ can be obtained by evaluating the nth derivative at $t = 0$. Inasmuch as our analysis is based on the generating function, we define Q by this relation. In addition, we define the function to be zero if any of the arguments is negative. The reason for extending the definition in this way is that it allows us to write summations without bothering to explicitly show the limits of summation.

$$Q(n,m,r) \equiv \begin{cases} 0 & \text{if } n,\ m, \text{ or } r < 0; \\ D^n \left(1+\frac{t}{1!}+\cdots+\frac{t^r}{r!}\right)^m \Big|_{t=0} & \text{otherwise.} \end{cases} \tag{1}$$

These last two equations show the generating functions for $Q(n,m,r)$. The recurrence relation that will be given here is not in terms of $Q(n,m,r)$, but rather uses a normalized version of this quantity, called $P(n,m,r)$. Whereas $Q(n,m,r)$ represents the *number* of ways that something can occur, $P(n,m,r)$ represents the *fraction* (or probability) of cases for which this occurs.

$P(n,m,r)-$ Fraction of words containing n letters, drawn from an m-letter alphabet, for which no letter occurs more than r times.

As before, we define the function over all the integers by setting it to zero for negative arguments. We also need to define what happens when $m = n = 0$ in a way consistent with the recurrence formula we will be deriving.

$$P(n,m,r) \equiv \begin{cases} 0 & \text{if } n,\ m \text{ or } r < 0; \\ 1 & \text{if } m = n = 0 \text{ and } r \ge 0; \\ \dfrac{Q(n,m,r)}{m^n} & \text{otherwise.} \end{cases}$$

The function we call F is the weighting factor in the recurrence relation. A considerable part of the analysis that follows is given to investigation of the properties of this function. F has four arguments, all of them integers. The definition is adjusted so that $F(k;n,m,r)=0$ whenever F is the weighting factor of something that is itself equal to zero. The function is defined by:

$$F(k;n,m,r) \equiv \begin{cases} 0 & \text{if } n,\ m \text{ or } r<0, \text{ or } rm<n; \\ \dfrac{n!}{m^n (r!)^m} & \text{if } m=k \text{ and } n=kr; \\ \dbinom{m}{k} \dfrac{[n]_{kr}}{m^{kr}(r!)^k} \left(\dfrac{m-k}{m}\right)^{n-kr} & \text{otherwise.} \end{cases} \tag{2}$$

(The second case above is the same as the last one, if we make the interpretation $0^0=1$.)

Lemma 1. *Let $r \geq 1$. Then $P(n,m,r)$ satisfies the following recurrence relation:*

$$P(n,m,r) = \sum_k F(k;n,m,r) \cdot P(n-kr, m-k, r-1). \tag{3}$$

Proof. Use the definition (1) of Q, along with the binomial theorem and Leibniz's rule for the nth derivative of a product. There results

$$\begin{aligned} Q(n,m,r) &= D^n \left(1+\frac{t}{1!}+\cdots+\frac{t^r}{r!}\right)^m \Bigg|_{t=0} \\ &= D^n \sum_k \binom{m}{k}\left(\frac{t^r}{r!}\right)^k \left(1+\frac{t}{1!}+\cdots+\frac{t^{r-1}}{(r-1)!}\right)^{m-k} \Bigg|_{t=0} \\ &= \sum_{k,l} \binom{m}{k}\binom{n}{l} \frac{D^l(t^{kr})}{(r!)^k} D^{n-l} \left(1+\frac{t}{1!}+\cdots+\frac{t^{r-1}}{(r-1)!}\right)^{m-k} \Bigg|_{t=0} . \end{aligned}$$

At $t=0$, the factor $D^l(t^{kr})=0$ if $l \neq kr$, and $D^{kr}(t^{kr})=(kr)!$. Moreover, $D^{n-l}(1+(t/1!)+\cdots+(t^{r-1}/(r-1)!))^{m-k}|_{t=0} = Q(n-l,m-k,r-1)$ from the definition of Q. Using this in the last equation gives

$$\begin{aligned} Q(n,m,r) &= \sum_k \binom{m}{k}\binom{n}{kr}\frac{(kr)!}{(r!)^k} Q(n-kr,m-k,r-1) \\ &= \sum_k \binom{m}{k}\frac{[n]_{kr}}{(r!)^k} Q(n-kr,m-k,r-1). \end{aligned}$$

Although this sum is formally an infinite series in k, in fact only a finite number of terms are nonzero. If $k<0$ or $k>m$, then the factor $\binom{m}{k}$ is equal to zero, and if $n<kr$, then $[n]_{kr}=0$. Note that, if $n-kr=m-k=0$, then $Q(n-kr,m-k,r-1)=1$, so that the term corresponding to the last expression above reduces to $[n]_{kr}/(r!)^k = n!/(r!)^m$. When we next divide by m^n, this case is in accord with the second line of the definition (2).

Now substitute the relation $Q(n,m,r) = m^n P(n,m,r)$ into both sides of the last equation. After dividing through by m^n, we get

$$P(n,m,r) = \sum_k \binom{m}{k} \frac{[n]_{kr}}{m^{kr}(r!)^k} \left(\frac{m-k}{m}\right)^{n-kr} P(n-kr, m-k, r-1),$$

which is equivalent to (3). ∎

There are several obvious properties satisfied by the function F that will be used in subsequent discussion. We list them here without proof.

Note first of all that $F(k;n,m,r) \geq 0$ everywhere. Furthermore, for fixed n, m, and r, the set of k for which $F(k;n,m,r) > 0$ is a subsequence of the integers, without any gaps. A similar statement holds if k, n, and m are fixed, and r varies. Finally, if $k = 0$ and $n, m, r \geq 0$, then by inspection all the factors in (2) are equal to one, so that $F(0;n,m,r) = 1$ as well.

In the next section, we will derive inequalities satisfied by the recurrence, and later use these to obtain a limit theorem concerning the growth of the expected maximum queue length when n and m grow large.

5 Inequalities

The recurrence that is shown in Lemma 1 appears too complicated to solve exactly in an efficient manner. In this section, we show that the terms of the recurrence obey certain inequalities. This will allow us to approximate the factor $P(n-kr, m-k, r-1)$ by an expression (namely, $P(n,m,r-1)$) that can be taken outside the summation in the recurrence formula. The inequalities also permit us to bound the error that results when we drop most of the terms of the summation. Together, these two simplifications lead to an efficient algorithm for estimating the probability distribution of the maximum queue length.

Before stating the inequalities, we recall a pair of definitions we will be using. A sequence of numbers $\{a_0, a_1, \ldots a_n\}$ is called *unimodal* if it has a single maximum; i.e., if there exists an index M such that $a_0 \leq a_1 \leq \cdots \leq a_{M-1} \leq a_M \geq a_{M+1} \geq \cdots \geq a_n$.

A sequence of nonnegative numbers $\{a_0, a_1, \ldots a_n\}$ is called *log-concave* if

$$a_j^2 \geq a_{j-1}a_{j+1} \quad \text{for } j = 1, \ldots, n-1.$$

Notice that a log-concave sequence is necessarily unimodal. This is because a non-unimodal sequence must contain a value a_j that is a local minimum, with a_j strictly less than at least one of its neighboring elements, and the sequence does not satisfy the definition of being log-concave for that a_j. For our purposes, the most important property of a log-concave sequence is that, if the sequence is also decreasing, then it decreases faster than a geometric sequence—i.e., super-exponentially. To see this, assume that all the a_j in the definition are positive, and $a_0 \geq a_1 \geq \cdots \geq a_n$. Then, applying the above definition for $j = 1, 2, \ldots, n-1$:

$$\frac{a_1}{a_0} \geq \frac{a_2}{a_1} \geq \cdots \geq \frac{a_n}{a_{n-1}}.$$

If we replace all the inequalities in this expression by equal signs, then we have a geometric sequence.

The proofs of the inequalities are rather lengthy and technical. For this reason, we first state them, and follow with the proofs. The reader may wish to skip over the proofs on a first reading. Among the results shown here, only Propositions 4, 7, and 9 are used later. Proposition 4 is referenced in the next section, which describes the computer algorithm based on these inequalities. It allows the factor $P(n-kr, m-k, r-1)$ in (3) to be replaced by $P(n, m, r-1)$; the latter is independent of k, and so can be taken outside the summation. Propositions 7 and 9 pertain to the other factor in (3), $F(k; n, m, r)$. Roughly speaking, they state that $F(k; n, m, r)$ decreases rapidly with increasing k and r. These two results are used in the proofs of the limit theorem in Section 7. Each proposition is preceded by one or two lemmas used in the proof of the proposition. The lemmas themselves are not used outside of this section.

Lemma 2. *Let $n \geq 1$. Then $P(n, m, r) \leq P(n-1, m, r)$.*

Lemma 3. *Let $r \geq 0$, $n \geq r$ and $m \geq 1$. Then $P(n, m, r) \leq P(n-r, m-1, r)$.*

Proposition 4. *Given $r \geq 1$, then for all k in the range $0 \leq k \leq \min(m, \lfloor n/r \rfloor)$:*

$$P(n, m, r-1) \leq P(n-kr, m-k, r-1).$$

Lemma 5. *Let $r \geq 1$. Then $F(k; n, m, r) \leq F(k; n, m, r-1)$.*

Lemma 6. *As a function of r, the sequence of values $F(k; n, m, r)$ is log-concave.*

Proposition 7. *Suppose n, m, and k are held fixed, and r varies. Then the set of values $F(k; n, m, r)$ decays to zero at a super-exponential rate.*

Lemma 8. *Suppose n, m, and r are held fixed, and k varies. Then, as a function of k, the sequence $F(k; n, m, r)$ is unimodal. In fact, given n and m, the sequence is log-concave everywhere with the exception of at most one pair (k, r), the exception occurring when $k = m$ and $r = n/m$.*

Proposition 9. *Suppose n, m, and r are such that $0 < F(1; n, m, r) < 1$, and that $r \neq n/m$. Then, as a function of k, $F(k; n, m, r)$ decays monotonically to zero at a super-exponential rate.*

Now we come to the proofs. The proofs of the first two lemmas do not use the recurrence relation, but rather work directly on the properties of the sets whose cardinalities are being compared.

Suppose A and B are two finite sets, and there is a map $f\colon A \to B$. By the definition of a map, for each $x \in A$, the map f associates exactly one $y = f(x) \in B$. Using standard notation, let $f^{-1}(y)$ be the set of all $x \in A$ for which $f(x) = y$. Then $y_1 \neq y_2 \Rightarrow f^{-1}(y_1) \cap f^{-1}(y_2) = \emptyset$, and $\bigcup_{y\in B} f^{-1}(y) = A$. This leads to the following estimate of the size of A relative to the size of B:

$$|A| \leq |B| \cdot \max_{y \in B} |f^{-1}(y)|.$$

In order to make use of this argument in the next two proofs, it is worthwhile to have a name for the set containing $Q(n, m, r)$ words that was described earlier.

$W(n, m, r)-$ Set of words containing n letters, drawn from an m-letter alphabet, for which no letter occurs more than r times.

As might be expected, we define $W(n, m, r)$ to be the null set if n, m, or $r < 0$. If $n = m = 0$, and $r \geq 0$, then $W(n, m, r)$ is a set whose sole member is the "empty word". In all cases, we have $Q(n, m, r) = |W(n, m, r)|$.

Proof of Lemma 2. The lemma states that if we increase the number of processors, and hold the number of memories constant, then the expected maximum number of references per memory goes up rather than down.

By representing P in terms of Q, it can be seen that the statement to be proved is equivalent to $m^{-n}Q(n, m, r) \leq m^{-(n-1)}Q(n-1, m, r)$, or $Q(n, m, r) \leq m \cdot Q(n-1, m, r)$. By the remarks made above, it suffices to find a map $f\colon W(n, m, r) \to W(n-1, m, r)$ for which $|f^{-1}(y)| \leq m$ for all $y \in W(n-1, m, r)$. We let f be the map that truncates the last letter from the word. This letter can be chosen in at most m ways, so $|f^{-1}(y)| \leq m$. ∎

Proof of Lemma 3. This is equivalent to $m^{-n}Q(n, m, r) \leq (m-1)^{-(n-r)}Q(n-r, m-1, r)$, or

$$Q(n, m, r) \leq \left(\frac{m}{m-1}\right)^{n-r} m^r Q(n-r, m-1, r).$$

It is sufficient to show a map $f\colon W(n, m, r) \to W(n-r, m-1, r)$ for which $|f^{-1}(y)| \leq \big(m/(m-1)\big)^{n-r} m^r$ for all $y \in W(n-r, m-1, r)$.

Let $x = a_1 \ldots a_r \in W(n, m, r)$, and let $f(x) = b_1 \ldots b_{(n-r)} \in W(n-r, m-1, r)$. By the definitions of their respective sets, the letters a_i are drawn from an m-letter alphabet, and the b_i are drawn from an $(m-1)$-letter alphabet. Without loss of generality, assume these two alphabets are identical except for the letter Z, which is present only in the larger alphabet. Call the first $n-r$ letters $a_1 \ldots a_{(n-r)}$ the *left part* of x, and the remaining letters $a_{(n-r+1)} \ldots a_n$ the *right part*. Assume the left part of x contains j occurrences of the letter Z, and the right part contains k occurrences of Z. Since $x \in W(n,m,r)$, then by definition $j+k \leq r$.

The map f can be viewed as the composition of two maps. The first map takes the k Z's in the right part and moves them to the end of the word, without changing the relative order of the other $r-k$ letters in the right part of x. The second map then takes the j Z's in the left part of y, and exchanges them with the first j letters of the right part of y (none of which is a Z), without changing the relative order of the letters moved. $f(y)$ then consists of the first $n-r$ letters of the following word:

$$\overbrace{a'_1 \ldots a'_{(n-r)}}^{\text{no } Z\text{'s}} \underbrace{Z \ldots Z}_{j \text{ times}} \overbrace{a'_{(n-r+j+1)} \cdots a'_{(n-k)}}^{\text{no } Z\text{'s}} \underbrace{Z \ldots Z}_{k \text{ times}}.$$

For a given $y \in W(n-r, m-1, r)$, the k Z's that are moved by f can be chosen in $\binom{r}{k}$ ways. The j Z's on the left can be chosen in $\binom{n-r}{j}$ ways. There are an additional $r-j-k$ letters on the right that are not Z's; each can be chosen independently from the remaining $m-1$ letters, so they contribute a factor $\leq (m-1)^{r-j-k}$ (this is an overestimate, since certain choices of these letters may cause a single letter to occur more than r times). To find the total number of elements in the inverse image $f^{-1}(y)$, we sum over all combinations of j and k, using the binomial theorem.

$$\begin{aligned}
|h^{-1}(y)| &\leq \sum_{j=0}^{r} \binom{n-r}{j} \sum_{k=0}^{r-j} \binom{r}{k} (m-1)^{r-j-k} \\
&\leq (m-1)^r \sum_j \binom{n-r}{j} \left((m-1)^{-1}\right)^j \sum_k \binom{r}{k} \left((m-1)^{-1}\right)^k \\
&= (m-1)^r \left(1 + \frac{1}{m-1}\right)^{(n-r)+r} = m^r \left(\frac{m}{m-1}\right)^{n-r}.
\end{aligned}$$

This satisfies the condition given at the beginning of the proof. ∎

Proof of Proposition 4. For $k=0$, there is nothing to prove. If $k=1$, then by the two previous lemmas, $P(n,m,r-1) \leq P(n-1,m,r-1) \leq P(n-1-(r-1), m-1, r-1) = P(n-r, m-1, r-1)$. For $k>1$, the result follows by induction. ∎

Proof of Lemma 5. If $r \geq 1$, and $F(k;n,m,r-1) = 0$, then it follows immediately from the definition of F that $F(k;n,m,r) = 0$ as well. Assume then that $F(k;n,m,r-1) > 0$. Since the proposition follows trivially if $F(k;n,m,r) = 0$, assume $F(k;n,m,r) > 0$ also. Using (2), and omitting factors that are independent of r (since their ratio is one), we have:

$$\frac{F(k;n,m,r)}{F(k;n,m,r-1)} = \frac{m^{k(r-1)}\left((r-1)!\right)^k}{m^{kr}(r!)^k} \frac{[n]_{kr}}{[n]_{k(r-1)}} \left(\frac{m-k}{m}\right)^{[(n-kr)-(n-k(r-1))]}$$

$$= \frac{[n-k(r-1)]_k}{m^k r^k} \left(\frac{m-k}{m}\right)^k \leq \left(\frac{n}{rm}\right)^k.$$

We are assuming $F(k;n,r,m) > 0$. However, by the way in which F was defined, this can occur only if $n \leq rm$, so that the last expression above is ≤ 1. This completes the proof. ∎

Proof of Lemma 6. We must show

$$F(k;n,m,r)^2 \geq F(k;n,m,r-1)\, F(k;n,m,r+1).$$

We have already remarked that $F(k;n,m,r) \geq 0$ for any set of arguments. If $F(k;n,m,r-1) = 0$ or $F(k;n,m,r+1) = 0$, then the statement is trivially true. If not, divide through the above formula by its right hand side. We find

$$\begin{aligned} \frac{F(k;n,m,r)^2}{F(k;n,m,r-1)\,F(k;n,m,r+1)} &= \frac{m^{k(r-1)} m^{k(r+1)}}{(m^{kr})^2} \left(\frac{m-k}{m}\right)^{2kr-k(r-1)-k(r+1)} \\ &\quad \times \left(\frac{(r-1)!\,(r+1)!}{(r!)^2}\right)^k \frac{([n]_{kr})^2}{[n]_{(k-1)r}[n]_{(k+1)r}} \\ &= \left(\frac{r+1}{r}\right)^k \frac{(n-k(r-1))!\,(n-k(r+1))!}{\big((n-kr)!\big)^2} \geq 1. \quad ∎ \end{aligned}$$

Proof of Proposition 7. Immediate from the two previous lemmas, and the earlier remark that a decreasing log-concave sequence decays super-exponentially. ∎

Proof of Lemma 8. Because of its length, the proof is given in the appendix at the end of this paper. ∎

Proof of Proposition 9. The conditions imply that n, m, and r are all nonnegative. Direct substitution into (2) with $k = 0$ shows that $F(0;n,m,r) = 1$. By definition, $F(k;n,m,r) = 0$ if $k < 0$. Then, by the previous lemma, for the given n, m, and r, the function F attains its maximum at $k = 0$ and decreases for all higher values of k. The decay is super-exponential because the sequence is log-concave. ∎

6 Computer Algorithm

The inequalities that were derived in the previous section serve as the basis of a computer algorithm to determine upper and lower bounds of the maximum queue length as a function of m and n. In this section, we very briefly describe how this is done.

Note first that the basic recurrence relation (3) implies the inequality $P(n,m,r) \geq \sum_k F(k;n,m,r) \cdot P_{\text{low}}(n-kr, m-k, r-1)$, where P_{low} is any lower bound of the function P. In particular, using Proposition 4, we have $P(n,m,r) \geq \sum_k F(k;n,m,r) \cdot P(n,m,r-1) = P(n,m,r-1) \cdot \sum_k F(k;n,m,r)$. This equation can be turned around to get an upper bound for $P(n,m,r-1)$ in terms of an upper bound for $P(n,m,r)$:

$$P(n,m,r-1) \leq P(n,m,r) \cdot \left(\sum_k F(k;n,m,r)\right)^{-1} \leq P_{\text{high}}(n,m,r) \cdot \left(\sum_{k=0}^{k_0} F(k;n,m,r)\right)^{-1}, \qquad (4)$$

where P_{high} is any given upper bound of P, and $k_0 \geq 0$ is any nonnegative index at which to terminate the summation. We have written a computer program that uses this relation to find an upper bound on the probability distribution function $P(n,m,r)$. The stopping index k_0 is determined by terminating the summation loop when $F(k;n,m,r)$ becomes smaller than some threshold value.

To get a lower bound, we use the following expression based on (8) (whose simple derivation is given later):

$$P(n,m,r-1) \geq P_{\text{low}}(n,m,r) - \sum_{k\geq 1} F(k;n,m,r) \cdot P_{\text{high}}(n-kr, m-k, r-1).$$

For the estimate $P_{\text{high}}(n-kr, m-k, r-1)$, we can simply use the obvious relation $P(n,m,r) \leq 1$. This is what is done in the following section, when we find the asymptotic behavior of $P(n,m,r)$. In the computer program, this estimate gives rather crude results. To get tighter bounds, we use a variation of (4) in order to find $P_{\text{high}}(n-kr, m-r, r-1)$.

7 Asymptotic Behavior

We now investigate the asymptotic behavior of the maximum queue length. We do so under the conditions that n and m grow to infinity while maintaining a constant ratio that we call $\lambda \equiv n/m$. We obtain the following results.

First, if $\lambda > 0$, then the expected maximum queue length L_{avg} grows without bound when n and m grow in this fashion, and $L_{\text{avg}} \cdot (\log m / \log\log m)^{-1} \to 1$ as $m \to \infty$ for any value of λ. Secondly, L_{avg} can be found by looking at the weighting function evaluated at $k = 1$. In particular, L_{avg} is close to the integer r_1 for which $F(1;n,m,r_1-1) \geq (r_1-1)^{1/2}$ and $F(1;n,m,r_1) < r_1^{1/2}$. Finally, the convergence of L_{avg} to r_1 is almost certain in the following sense. Given λ and given $\epsilon > 0$, there exists an m_0 such that the probability that the maximum queue length differs from r_1 by more than one is less than ϵ whenever $m \geq m_0$. The value m_0 depends on both λ and ϵ.

As we develop the proofs of these statements, we will twice briefly digress to show how they relate to the graphs in Figures 1–4 that were discussed earlier. In particular, we will see that the growth rate of the maximum queue length, and the fact that this value is concentrated at only a few values, is reflected in the propositions to be demonstrated in this section.

We start by defining the values r_0 and r_1 around which the maximum queue length clusters.

$$r_0 = r_1 - 1, \quad \text{where}$$

$$r_1 = \min_{r \geq \lambda} \left\{ r \colon F(1;n,m,r) \leq \sqrt{r} \right\}.$$

Since $F(1;n,m,r) = 0$ if $r > n$, and $r_1 \geq \lambda$, the value r_1 is well-defined.

Before stating the theorem, we need to establish several lemmas.

Lemma 10. *Let $\lambda > 0$ be a rational number. Let n and m grow to infinity while maintaining a constant ratio $\lambda = n/m$. Let r_0 and r_1 be defined as above. Then, as m goes to infinity, r_0 and r_1 also go to infinity.*

Proof. Assume the contrary. Suppose r_1 were bounded, say by $r_1 \le R$. Using the definition (2) of F, evaluated at $k = 1$, we have

$$F(1; n, m, r_1) = m \frac{[n]_{r_1}}{m^{r_1} r_1!} \left(\frac{m-1}{m}\right)^{(n-r_1)} \ge m \frac{1}{R!} \frac{[n]_{r_1}}{m^{r_1}} \left(\frac{m-1}{m}\right)^{\lambda m}.$$

We are assuming that λ is constant and r_1 is bounded. This implies that, for large m:

$$\frac{[n]_{r_1}}{m^{r_1}} \to \left(\frac{n}{m}\right)^{r_1} = \lambda^{r_1}; \qquad \left(\frac{m-1}{m}\right)^{\lambda m} \to e^{-\lambda}.$$

Using this in the above formula, we have for large m:

$$F(1; n, m, r_1) \ge \theta m \frac{1}{R!} e^{-\lambda} \cdot \min\{1, \lambda^R\},$$

where $\theta = 1 + o(1)$. The expression on the right is unbounded as $m \to \infty$. By definition, however, $F(1; n, m, r_1) < r_1^{1/2} \le R^{1/2}$, a contradiction. Therefore, r_1 must be unbounded, and so too must $r_0 = r_1 - 1$. ∎

Lemma 11. *With the same assumptions as in the previous lemma, the ratio*

$$\frac{F(1; n, m, r_1)}{F(1; n, m, r_0)} \to 0 \qquad \text{as } m \to \infty.$$

Proof. We substitute into the expression shown above for $F(1; n, m, r)$:

$$\begin{aligned} \frac{F(1; n, m, r+1)}{F(1; n, m, r)} &= \frac{r!}{(r+1)!} \frac{m^r}{m^{r+1}} \left(\frac{m}{m-1}\right)^{(r+1)-r} \frac{[n]_{r+1}}{[n]_r} \\ &= \frac{1}{r+1} \cdot \frac{n-r}{m-1} = \frac{\lambda}{r+1} \cdot \frac{m - r/\lambda}{m-1} \to 0 \qquad \text{as } r \to \infty. \end{aligned} \tag{5}$$

Letting $r = r_0$, and using the previous lemma, gives this lemma. ∎

Lemma 12. *Under the same conditions, $r_0, r_1 = o(\log m)$* $\qquad$ as $m \to \infty$.

Proof. We have

$$F(1; n, m, r_0) = m \frac{[n]_{r_0}}{m^{r_0} r_0!} \left(\frac{m-1}{m}\right)^{n-r_0} \le m \frac{\lambda^{r_0}}{r_0!} \le m \left(\frac{\lambda e}{r_0}\right)^{r_0}, \tag{6}$$

where the last inequality follows from Stirling's formula. Using the fact that $F(1; n, m, r_0) > r_0^{1/2} \ge 1$ by definition, we can take logarithms and rearrange, obtaining:

$$r_0 \le \frac{\log m}{\log r_0 - \log \lambda - 1}. \tag{7}$$

Since $r_0 \to \infty$ as $m \to \infty$, the denominator on the right is unbounded, which proves the statement for r_0. It then follows that $r_1 = o(\log m)$ as well. ∎

We now sharpen these results to find the asymptotic behavior of r_0 and r_1 as $m \to \infty$. We have by definition of r_1 that $\sqrt{r}_1 \geq F(1; n, m, r_1)$. We expand $F(1; n, m, r_1)$ in the same way as was done in (6). We multiply through by $r_1!$ and take logarithms, obtaining

$$\frac{1}{2}\log r_1 + \big(1 + o(1)\big) r_1 \log r_1 = \big(1 + o(1)\big) r_1 \log r_1 \geq \log m + \log\left(\frac{[n]_{r_1}}{m^{r_1}}\left(\frac{m-1}{m}\right)^{n-r_1}\right) = \big(1 + o(1)\big)\log m.$$

We have used a weakened version of Stirling's formula, that $\log k! = k \log k\big(1 + o(1)\big)$. We now divide by $\log r_1 = O(\log\log m)$, to get

$$r_1 \geq \big(1 + o(1)\big)\frac{\log m}{\log r_1} \geq \big(1 + o(1)\big)\frac{\log m}{\log\log m}.$$

Since $r_0 = r_1 + 1$, this shows that $r_0 \geq \big(1 + o(1)\big)\log m / \log\log m$ as well, which implies that $\log r_0 \geq \big(1 + o(1)\big)\log\log m$. Furthermore, we have by definition that $\sqrt{r}_0 \leq F(1; n, m, r_0)$. This means that we can repeat the same argument as above, with r_0 substituted for r_1, and with the sense of the inequality reversed. The result is

$$r_0 \leq \big(1 + o(1)\big)\frac{\log m}{\log r_0} \leq \big(1 + o(1)\big)\frac{\log m}{\log\log m}.$$

Putting this together, we have proved the following.

Proposition 13. *With the same assumptions as made previously, for any postive rational λ,*

$$r_0, r_1 = \big(1 + o(1)\big)\frac{\log m}{\log\log m} \qquad \text{as } m \to \infty.$$

The results of this propostion can be compared against the graphs shown in Figure 1. As mentioned before, these graphs are plotted on a semilog scale. According to the lemma, the slopes of the graphs should be decreasing as the independent variable m increases. This effect can be seen for the two higher curves, corresponding to $\lambda = 1.0$ and $\lambda = 4.0$, but the effect is very small, especially for the higher values of m. This is partly because $(\log m)$-growth is not too much different from $(\log m/\log\log m)$-growth for this range of numbers. We have also not examined how fast r_0 and r_1 converge to the asymptotic value given in the proposition, but it should be clear from the proof that some of the approximations are quite crude for practical values of m and n.

It should be noted that the same technique used to prove the previous proposition can be used to show that

$$\Gamma^{-1}(m) = \big(1 + o(1)\big)\frac{\log m}{\log\log m}$$

for large m. The expression on the left was the one obtained by Gonnet [1983] when studying an equivalent problem connected with hash queues. The equation expresses the fact that, to the order of approximation shown, the two expressions are asymptotically equal.

The limit theorem describes where the probability is concentrated. Since $P(n, m, r)$ is a cumulative probability distribution, the difference $P(n, m, r) - P(n, m, r-1)$ is the probability that the maximum number of references to any one memory is exactly r. ¿From Lemma 1, and the fact that at $k = 0$ the weighting factor $F(0; n, m, r) = 1$, we get an explicit formula for its value:

$$\delta P(n, m, r) \equiv P(n, m, r) - P(n, m, r-1) = \sum_{k \geq 1} F(k; n, m, r) \cdot P(n - kr, m - k, r - 1). \tag{8}$$

We are now ready to prove the main result of this section.

Theorem 14. *Let $\lambda > 0$ be a rational number. Let $n, m \to \infty$ at a constant ratio $n/m = \lambda$. Let r_0 and r_1 be defined as in the previous lemmas. Then*

a) Under these conditions, $L_{\text{avg}} \to \infty$, where L_{avg} is the expected value of the maximum queue length.

b) For any $\lambda > 0$:

$$\lim \sum_{r < r_0, r > r_1} \delta P(n, m, r) = 0.$$

Proof. If $r_0 \to \infty$, then it follows from (b) that the expected value of the maximum queue length becomes infinite as well, so that (a) would then be a consequence of (b). We have already shown in the lemmas that $r_0 \to \infty$, so it remains to show that the growth in the probability distribution $P(n, m, r)$ becomes concentrated around r_0 and r_1.

We look first at $r < r_0$. The proof follows easily from (4), taking as upper bound $P_{\text{high}}(n, m, r_0) = 1$:

$$\begin{aligned} P(n, m, r_0 - 1) &\leq \left(\sum_k F(k; n, m, r_0) \right)^{-1} \leq F(1; n, m, r_0)^{-1} \\ &\leq \frac{1}{\sqrt{r_0}} \to 0 \qquad \text{as } r_0 \to \infty. \end{aligned}$$

Since $r_0 \to \infty$ as $m \to \infty$, this proves half of (b):

$$\sum_{r < r_0} \delta P(n, m, r) = P(n, m, r_0 - 1) \to 0 \qquad \text{as } m \to \infty.$$

Next we look at $r > r_1$. It was shown previously, in Proposition 7 and Proposition 9, that the weighting function F decays super-exponentially with both k and r. This allows us to estimate $F(k; n, m, r)$ for $r > r_1$.

We have first of all that $F(1;n,m,r_1) \le r_1^{1/2}$ by definition. Then, using (5):

$$F(1;n,m,r+1) = \frac{\lambda}{r_1+1}\frac{m-r_1/\lambda}{m-1}F(1;n,m,r_1) \le \frac{\lambda}{r_1}F(1;n,m,r_1) \le \frac{\lambda}{\sqrt{r_1}}.$$

Since F decays super-exponentially with r, this implies, for $j \ge 1$,

$$F(1;n,m,r_1+j) \le \left(\frac{\lambda}{r_1}\right)^{j-1}\frac{\lambda}{\sqrt{r_1}} \le \left(\frac{\lambda}{\sqrt{r_1}}\right)^{j}.$$

F also decays super-exponentially with k. Recognizing that $F(0;n,m,r_1+j)=1$, and using what what just shown:

$$F(k;n,m,r_1+j) \le F(1;n,m,r_1+j)^k \le \left(\frac{\lambda}{\sqrt{r_1}}\right)^{jk}.$$

We now use all this along with (8), taking as upper bound $P(n-kr,m-k,r-1) \le 1$.

$$\begin{aligned}
\sum_{r>r_1}\delta P(n,m,r) &= \sum_{r>r_1}\sum_{k\ge 1}F(k;n,m,r)\cdot P(n-kr,m-k,r-1)\\
&\le \sum_{j\ge 1}\sum_{k\ge 1}\left(\frac{\lambda}{\sqrt{r_1}}\right)^{jk} = \sum_{j\ge 1}\left(\frac{\lambda}{\sqrt{r_1}}\right)^{j}\frac{1}{1-(\lambda/\sqrt{r_1})^j}\\
&\le \frac{1}{1-\lambda/\sqrt{r_1}}\sum_{j\ge 1}\left(\frac{\lambda}{\sqrt{r_1}}\right)^{j}\\
&= \frac{\lambda}{\sqrt{r_1}}\cdot\left(\frac{1}{1-\lambda/\sqrt{r_1}}\right)^2 \to 0 \qquad \text{as } r_1 \to \infty.
\end{aligned}$$

Since $r_1 \to \infty$ as $m \to \infty$, this completes the proof. ∎

Part (b) of this last theorem relates to the graphs shown in Figures 2–4. These graphs show the probability that the maximum queue length is equal to a given r as a function of m for constant λ. According to the theorem, one would expect the curves to have sharp peaks, with ever-narrower shape as m increases. The first of these effects is in fact evident from the graphs. It may be especially apparent from the bottom curves in these three figures that the probability is concentrated around a few values of r for each value of m. It is not so clear from these figures that the curves are narrower for large m than they are for small m. The detailed numeric results, which we have printed to six significant digits, do indicate that this occurs, but the effect is very small even for the wide range of values of m shown here.

In the form just given, one has to evaluate $F(1;n,m,r)$ at discrete values of r to get an approximation to L_{avg}. It is often easier to find the root of a continuous function than it is to test a discrete function at different points. Since $F(1;n,m,r_1)$ can be approximated by continuous functions, we get the following result.

Corollary 15. *Given the assumptions of the previous theorem, the same results hold if $\tilde{r}_0$ and $\tilde{r}_1$ are used in place of r_0 and r_1, respectively, where $\tilde{r}_0 = \lfloor r_a \rfloor$ and $\tilde{r}_1 = \lceil r_a \rceil$, and where r_a is defined by*

$$r_a \equiv \min \left\{ r \geq \lceil \lambda \rceil : \frac{1}{\sqrt{2\pi}} \frac{m}{r} \left(\frac{\lambda e}{r} \right)^r e^{-\lambda} \leq 1 \right\}. \tag{9}$$

Proof. We first remark that for fixed m and λ, the above expression is continuous and clearly goes to zero as $r \to \infty$, so r_a is well-defined. We have shown previously that $r_1 \to \infty$ as $m \to \infty$, and that $r_1 = o(\log m)$, so that we have the limits

$$\frac{[n]_{r_1}}{m^{r_1}} \to \left(\frac{n}{m} \right)^{r_1} = \lambda^{r_1}, \qquad \left(\frac{m-1}{m} \right)^{n-r_1} = \left(\frac{m-1}{m} \right)^{\lambda m - r_1} \to e^{-\lambda}$$

as $m \to \infty$. These were obvious when r_1 was assumed bounded. We still have $\big((m-1)/m\big)^{\lambda m} \to e^{-\lambda}$ since λ is constant. To show that the other factors converge to one, we use the formula $\log(1+x) = x + o(x)$ as $x \to 0$. Then, since $r_1 = o(\log m)$:

$$\log \left(\frac{[n]_{r_1}}{m^{r_1}} \lambda^{-r_1} \right) = \log \frac{[n]_{r_1}}{n^{r_1}} \geq \log \left(\frac{n - r_1}{n} \right)^{r_1} = r_1 \log \frac{n - r_1}{n} \to \frac{-r_1^2}{\lambda m} \to 0;$$

$$\log \left(\frac{m}{m-1} \right)^{r_1} = r_1 \log \left(1 + \frac{1}{m-1} \right) \to \frac{r_1}{m-1} \to 0.$$

We also have Stirling's formula: $r_1! = \vartheta_{r_1} (2\pi r_1)^{1/2} (r_1/e)^{r_1}$, where $\vartheta_{r_1} \to 1$ as $r_1 \to \infty$. We use these limits to approximate $F(1; n, m, r_1)$. In the formula that follows, θ represents the correction factor to account for the approximations. We have just shown $\theta \to 1$ as $m \to \infty$.

$$\begin{aligned} F(1; n, m, r_1) &= m \frac{[n]_{r_1}}{m^{r_1} r_1!} \left(\frac{m-1}{m} \right)^{(n - r_1)} \\ &= \theta \frac{1}{\sqrt{2\pi}} \frac{m}{\sqrt{r_1}} \left(\frac{\lambda e}{r_1} \right)^{r_1} e^{-\lambda} \leq \sqrt{r_1}. \end{aligned}$$

The inequality is true by definition of r_1. If we divide through by $r_1^{1/2}$ and compare with (9), we find that $r_a \to r_1$, except for two differences. First, r_1 is defined to be an integer, whereas r_a need only be real. Secondly, there is the factor of θ. So instead of $r_1 \to r_a$, we have $r_1 = \lceil \theta r_a \rceil$. But even though $\theta \to 1$, if r_a is close to one, it may be that $\lceil \theta r_a \rceil \neq \lceil r_a \rceil$, so that we cannot just cite the previous theorem.

The corollary is still true, however. We defined r_1 as a minimum integer satisfying $F(1; n, m, r_1) < r_1^{1/2}$. In the proofs, the only properties we used of the square root function are that $x^{1/2} \to \infty$ and $x^{1/2}/x \to 0$. We could as easily have defined r_1 by the relation $F(1; n, m, r_1) < r_1^{\alpha}$ for any α in the range $0 < \alpha < 1$. The corresponding effect in (9) is to change the factor r^{-1} to $r^{-(\alpha + 1/2)}$.

Assume then that we have defined r_1' and r_1'' in the same way as r_1 was defined, but using $\alpha = 0.4$ and $\alpha = 0.6$ respectively instead of $\alpha = 0.5$. The limit theorem now holds for both r_1' and r_1''. For m sufficiently large, since $\theta \to 1$, we have $r_a^{0.9} < r_a/\theta < r_a^{1.1}$.

Because of this, and because of the way in which r_1' and r_1'' are defined, for large m we will always have $r_1' \leq \lceil r_a \rceil \leq r_1''$. Since the limit theorem is true for both r_1' and r_1'', it is true for r_a as defined above. ∎

8 Summary and Conclusions

This paper has presented a new method for analyzing the lengths of memory queues when the network is conflict-free. An algorithm based on this method efficiently determines upper and lower bounds of the queue length. We have also analyzed the asymptotic behavior.

Our analysis indicates that the strategy of using hashing to spread data across memory modules is a good one. If the size of the system is increased, while maintaining a constant ratio of numbers of processors to memories, then asymptotically, the slowdown in performance from the effect studied by this paper is $\Theta(\log m / \log\log m)$. For m and n less than 100,000, and λ between 0.25 and 4.0, the graphical data confirm this growth rate.

Although it is worthwhile to have bounds on the value desired, this cannot be considered a full solution to the problem. A drawback of the method shown here is that it does not allow us to sharpen the estimates in a convenient manner. This is not important for the memory performance problem studied here, but it could matter in other applications, if more precise estimates were needed.

This paper has looked only at systems where there are independent data paths between each processor and memory, such as across a crossbar network. A crossbar requires n^2 crosspoints to connect n processors to n memories, and it would probably be too expensive for large n. Consequently, it would be worthwhile to perform similar analysis on other network topologies, such as the perfect shuffle, where the size of the network is $O(n \log n)$. The problem is more complex with other network topologies, because there can be conflicts within the network as well as at the memories. Kruskal and Snir [1983] have in fact looked at this problem for the perfect shuffle with the same assumptions as were made here, with processors making independent accesses to memories and with equal probability to access any memory. These assumptions are not always valid, and it would be worthwhile to analyze the same problem under different assumptions, such as when several of the memories are more favored to be referenced. Pfister and Norton [1985] have shown that an omega network gets saturated under these conditions. Some of their results were based on simulation, and it may be possible to get numeric expressions for the effect they found by using methods like those used here.

Acknowledgments.

I am very grateful for the help given me by Harold Stone, my advisor. It was he who suggested

this problem to me, and he introduced me to the use of generating functions in analyzing such problems. Afterwards, he critically read the paper through its several revisions, and greatly helped to improve its readability.

Appendix

Proof of Lemma 8

Here we prove Lemma 8. Suppose that the variables n, m, and r are held fixed, and $r \neq n/m$. We must show that the function $F(k; n, m, r)$ is log-concave in the variable k.

Assuming k, n, m, and r are all ≥ 0, we first determine which values of k make $F(k; n, m, r) = 0$. ¿From the definition (2) of F, this occurs only when at least one of the factors $\binom{m}{k}$, $[n]_{kr}$, or $m - k$ is equal to zero, unless $m - k = n - kr = 0$, which is excluded by hypothesis. Apart from this exceptional case, the function is zero when $k < 0$, or $k \geq m$, or $kr > n$. Consequently, $F(k; n, m, r)$ is positive if and only if

$$k \geq 0, \qquad m - k \geq 1, \qquad \text{and} \quad n - kr \geq 0. \tag{10}$$

In addition, the recurrence is used only for values of $r \geq 1$.

It turns out that the proof we are about to give does not work for the case $m - k = n - kr = 0$, and this is why we assume $r \neq n/m$ in the statement of the lemma. It can be shown that $F(k; n, m, r)$ is unimodal with k for this case, although not necessarily log-concave. Since we do not need this result, however, we do not give the proof. Rather, we assume the arguments of the function satisfy (10), and show that F is then log-concave in k. We must show

$$F(k; n, m, r)^2 \geq F(k-1; n, m, r)F(k+1; n, m, r) \tag{11}$$

in those cases where both sides of the inequality are positive. Using (10) for $F(k-1; n, m, r)$ and $F(k + 1; n, m, r)$, we see that we can limit ourselves to k in the range

$$k \geq 1, \qquad m - k \geq 2, \qquad \text{and } n - kr \geq r. \tag{12}$$

Working with k in this range, divide (11) through by its right hand side, and call the resulting expression "X". We do not explicitly show the dependence of X on k, n, m, and r. We must show

$$X = \frac{F(k; n, m, r)^2}{F(k-1; n, m, r)F(k+1; n, m, r)} \geq 1.$$

We now use the definition (2) of $F(k;n,m,r)$ to get an expression for X:

$$
\begin{aligned}
X = {} & \left[\frac{m!}{k!\,(m-k)!} \frac{n!}{(n-kr)!\,m^{kr}(r!)^k} \left(\frac{m-k}{m}\right)^{n-kr} \right]^2 \\
& \times \frac{(k-1)!\,(m-k+1)!}{m!} \frac{(n-kr+r)!\,m^{kr-r}(r!)^{k-1}}{n!} \left(\frac{m}{m-k+1}\right)^{n-kr+r} \\
& \times \frac{(k+1)!\,(m-k-1)!}{m!} \frac{(n-kr-r)!\,m^{kr+r}(r!)^{k+1}}{n!} \left(\frac{m}{m-k-1}\right)^{n-kr-r} \\
= {} & \frac{(k+1)}{k} \frac{(m-k+1)}{(m-k)} \frac{(n-kr+r)!(n-kr-r)!}{[(n-kr)!]^2} \left(\frac{(m-k)^2}{(m-k)^2-1}\right)^{n-kr-r} \left(\frac{m-k}{m-k+1}\right)^{2r}.
\end{aligned}
$$

We want to show that this is always ≥ 1 for (k,n,m,r) satisfying the conditions shown above. The problem is that the dependence of X on these four variables is not at all simple. If we increase or decrease any one of them, then some factor increases and another decreases.

As a first step in separating the dependence of X on the variables, notice that m always occurs as part of the expression $m-k$, and n occurs as $n-kr$. The conditions (12) are also expressed in terms of $m-k$ and $n-kr$, so we make the substitutions $M = m-k \geq 2$ and $N = n-kr \geq r$. X then becomes:

$$
X = \frac{(k+1)}{k} \frac{(M+1)}{M} \frac{(N+r)!\,(N-r)!}{(N!)^2} \left(\frac{M^2}{M^2-1}\right)^{N-r} \left(\frac{M}{M+1}\right)^{2r}. \tag{13}
$$

We now look for the value of N, in terms of k, r, and M, which minimizes X. In particular, we see what happens to X when N is increased by one.

$$
\begin{aligned}
\frac{X|_{N+1}}{X|_N} &= \frac{(N+1+r)!\,(N+1-r)!}{[(N+1)!]^2} \frac{(N!)^2}{(N+r)!(N-r)!} \left(\frac{M^2}{M^2-1}\right) \\
&= \frac{(N+1)^2-r^2}{(N+1)^2} \frac{M^2}{M^2-1} \begin{cases} < 1 & \text{if } (N+1)/r < M; \\ = 1 & \text{if } (N+1)/r = M; \\ > 1 & \text{if } (N+1)/r > M. \end{cases}
\end{aligned}
$$

Thus, X is minimized if $N = rM-1$ or $N = rM$. Choose the latter, and plug into (13).

$$
X \geq \frac{(k+1)}{k} \frac{(M+1)}{M} \frac{[r(M+1)]!\,[r(M-1)]!}{[(rM)!]^2} \left(\frac{M^2}{M^2-1}\right)^{r(M-1)} \left(\frac{M}{M+1}\right)^{2r}. \tag{14}
$$

For $r = 1$, this becomes

$$
X \geq \frac{(k+1)}{k} \frac{(M+1)}{M} \frac{(M+1)}{M} \left(\frac{M^2}{M^2-1}\right)^{M-1} \left(\frac{M}{M+1}\right)^2 = \frac{k+1}{k} \left(\frac{M^2}{M^2-1}\right)^{M-1} > 1.
$$

Thus, the statement is proved for $r = 1$. Assume now that $r \geq 2$. We use the following version of Stirling's formula, a stronger form of which appears in Mitrinović [1970], pp. 181 ff.:

$$
\sqrt{2\pi n}\,n^n e^{-n} < n! < \sqrt{2\pi n}\,n^n e^{-n} \exp\frac{1}{12n}.
$$

When this formula is used in (14), the factors of the form $(2\pi)^{1/2}e^{-n}$ cancel in the numerator and denominator. In the first formula that now follows, all factors from (14) except the approximation to the factorial functions appear on the first line. We are left with the following:

$$\begin{aligned} X &> \frac{(k+1)}{k}\frac{(M+1)}{M}\left(\frac{M^2}{M^2-1}\right)^{r(M-1)}\left(\frac{M}{M+1}\right)^{2r} \\ &\times\left(\frac{r^2(M^2-1)}{r^2M^2}\right)^{r(M-1)+1/2}\left(\frac{r(M+1)}{rM}\right)^{2r}\exp\left(-\frac{2}{12rM}\right) \\ &> \frac{M+1}{M}\left(\frac{M^2-1}{M^2}\right)^{1/2}\exp\frac{-1}{6rM} = \left(\frac{M^4+2M^3-2M-1}{M^4}\right)^{1/2}\exp\frac{-1}{6rM}. \end{aligned} \tag{15}$$

By taking logarithms, and using the fact that $r \geq 2$ and $M \geq 2$, it is easily verified that this last expression is > 1. The argument of the exponential function is increasing in r, so its logarithm can be estimated by evaluating the argument at $r = 2$. The logarithm of the other factor is estimated by using the following inequality from Mitrinović [1970], p. 273:

$$0 < y < x \Rightarrow \log\frac{x}{y} > 2\frac{x-y}{x+y}.$$

Since $M \geq 2$, there are also the obvious relations $M + 1/2 < M^3/2$ and $M^2 - 1 < M^3/2$. Putting this in (15) gives

$$\begin{aligned} \log X &> \frac{1}{2}\times 2 \times \frac{2M^3-2M-1}{2M^4+2M^3-2M-1} - \frac{1}{6rM} \\ &> \frac{1}{M}\times\frac{M^3-M-1/2}{M^3+M^2-1} - \frac{1}{12M} \\ &> \frac{1}{M}\times\frac{M^3-M^3/2}{M^3+M^3/2} - \frac{1}{12M} = \frac{1}{4M} > 0. \end{aligned}$$

This shows that $X > 1$, which means that $F(k;n,m,r)$ is log-concave in the variable k. ∎

References

Ian F. Akyildiz
"Mean Value Analysis for Blocking Queueing Networks," *IEEE Trans. Soft. Eng.*, vol. 14, no. 4, April 1988, pp. 418–427.

Selim G. Akl
Parallel Sorting Algorithms, Academic Press, Orlando, FL, 1985.

Dwass, M.
"Note on More Birthday Surprises," *J. Combinatorial Theory*, vol. 7, 1969, pp. 258–261.

Flajolet, P.
"On the Performance Evaluation of Extendible Hashing and Trie Searching," *Acta Informatica*, vol. 20, 1983, pp. 345–369.

Gonnet, G. H.
"Expected Length of the Longest Probe Sequence in Hash Code Searching," *Journal of the ACM*, vol. 28, no. 2, 1983, pp. 289–304.

Johnson, N. L., and S. Kotz
Urn Models and Their Application, Wiley, New York, 1977.

Klamkin, M. S., and D. J. Newman
"Extensions of the Birthday Surprise," *J. Combinatorial Theory*, vol. 3, 1967, pp. 279–282.

Knuth, D. E.
Fundamental Algorithms, The Art of Computer Programming, vol I, 2nd ed., Addison-Wesley, Reading, MA, 1973.

Clyde P. Kruskal and Marc Snir
"The Performance of Multistage Interconnection Networks for Multiprocessors," *IEEE Trans. Comput.*, vol. C-32, no. 12, December 1983, pp. 1091–1098.

Mendelson, H.
"Note On Permutations with Limited Repetition ," *J. Combinatorial Theory, Series A*, vol. 30, 1981, pp. 351–353.

Mitrinović, D. S.
Analytic Inequalities, Springer-Verlag, New York, 1970.

Pfister, G. F. and V. A. Norton
"'Hot Spot' Contention and Combining in Multistage Interconnection Networks," 1985, pp. 790–797.