The Limited Performance Benefits of Migrating Active Processes for Load Sharing

Derek L. Eager

Department of Computational Science
University of Saskatchewan

Edward D. Lazowska and John Zahorjan

Department of Computer Science
University of Washington

Abstract

Load sharing in a distributed system is the process of transparently sharing workload among the nodes in the system to achieve improved performance. In *non-migratory* load sharing, jobs may not be transferred once they have commenced execution. In load sharing with *migration*, on the other hand, jobs in execution may be interrupted, moved to other nodes, and then resumed.

In this paper we examine the performance benefits offered by migratory load sharing beyond those offered by non-migratory load sharing. We show that while migratory load sharing can offer modest performance benefits under some fairly extreme conditions, there are *no* conditions under which migration yields *major* performance benefits.

CR Categories and Subject Descriptors: C.2.4 [Computer-Communication Networks]: Distributed Systems – network operating systems; D.4.8 [Operating Systems]: Performance – modelling and prediction

General Terms: Design, Performance

Additional Key Words and Phrases: load sharing, migration, queueing models, local area networks

1. Introduction

A distributed computer system consists of a collection of individual systems (nodes) that share resources. Distributed systems utilizing *load sharing* attempt to share processing power, and thus to improve system performance, by transparently transferring work between nodes.

In *adaptive* load sharing, transfer decisions are based on the current system state, rather than just on information about the average behavior of the system. Previous studies have shown that adaptive load sharing has the potential to greatly improve system performance over that obtained with no load sharing or with non-adaptive load sharing, and that much of this potential can be realized with quite simple policies [Livny & Melman 1982; Eager, Lazowska & Zahorjan 1986a]. This potential has prompted a number of studies of specific load sharing policies [Bryant & Finkel 1981; Barak & Shiloh 1985; Krueger & Finkel 1984; Wang & Morris 1985; Eager, Lazowska & Zahorjan 1986b; Hsu & Liu 1986; Lee & Towsley 1986; Leland & Ott 1986], as well as the implementation of various mechanisms to support load sharing [Hwang et al. 1982; Powell & Miller 1983; Barak & Litman 1984; Theimer, Lantz & Cheriton 1985; Bershad 1985; Hagmann 1986; Litzkow 1987; Nichols 1987].

Adaptive load sharing policies may be broadly classified into two categories depending on whether they perform *migration* or instead rely solely on *initial placement*. A load sharing policy relying solely on initial placement may only transfer a job from one node to another when that job first originates, i.e., prior to its entry into the multiprogramming mix at some node. If a load sharing policy utilizes migration, a job in execution may be interrupted, moved to another node, and then resumed.

Since from a systems point of view it can be difficult to provide migration of active processes – especially *efficient* migration – and since most existing systems do not do so, a natural and fundamental question is whether migration can offer significant performance benefits beyond those available through non-migratory load sharing policies, or whether initial placement alone is sufficient. It is easy to convince oneself that migration should provide some (perhaps major) performance improvement. Since in most systems the service demands of jobs are not known *a priori*, with initial placement jobs are assigned to nodes in ignorance of these demands. An initial distribution of jobs across nodes that appears balanced will therefore become imbalanced as shorter jobs complete and leave behind an uneven distribution of longer jobs. Migration allows such imbalances to be corrected.

Most past studies that have explicitly considered migration do not attempt to quantify its potential for improving performance, but simply assume that this potential is significant. A notable exception is the work of Leland and Ott [1986], who show, using trace-driven simulation, that migration can improve performance for the large

This material is based upon work supported by the National Science Foundation (Grants No. DCR-8352098, CCR-8619663 and CCR-8703049), the Naval Ocean Systems Center, U S WEST Advanced Technologies, the Washington Technology Center, Digital Equipment Corporation's External Research Program, and the Natural Sciences and Engineering Research Council of Canada. Part of this work was done while Zahorjan was on sabbatical leave at Laboratoire MASI, University Paris 6.

Authors' addresses: Derek L. Eager, Department of Computational Science, University of Saskatchewan, Saskatoon, Canada S7N 0W0; Edward D. Lazowska and John Zahorjan, Department of Computer Science FR-35, University of Washington, Seattle, WA 98195.

jobs that occur in their traces (the *hogs*, in their colorful terminology). Their results indicate an improvement of less than 25% for these jobs, however, and they indicate no significant potential for improving the *average* performance over all jobs. Some analytical support for this negative result is provided by Kruskal and Weiss [1985], but their models do not allow high variability in service demands, a situation in which one would expect the benefits of migration to be relatively significant.

In this paper we attempt to identify the potential performance benefits of migration through the application of simple analytic and simulation models. (Note that we focus here only on the use of migration as a tool for improving system performance. In practice, there might be other motivations for implementing migration.) We conclude that:

- There are likely *no* conditions under which migration could yield *major* performance improvements beyond those offered by non-migratory load sharing, particularly when viewed relative to the advantages of non-migratory load sharing over no loadsharing.
- Under some fairly extreme conditions, migration *can* offer *modest* additional performance improvements. These extreme conditions are characterized by high variability in *both* job service demands and the workload generation process.
- The benefits of migration are not limited by its cost, but rather by the inherent effectiveness of non-migratory load sharing. Whenever migration does offer significant potential benefits, the migration cost would likely be dwarfed by the (very large) service demands of the jobs that can be fruitfully migrated.
- Different job service demand distributions that match with respect to both mean and variance may yield quite different results concerning the benefits of migration, thus caution is needed when developing workload models for use in migration studies.

In Section 2 of this paper we study migration through the use of a simple "no arrivals" model that allows analytical solutions for most of the quantities of interest. Those issues that cannot be effectively studied using this model are treated in Section 3 using a more conventional open queueing model that is solved using simulation. Section 4 concludes the paper.

2. A "No Arrivals" Model and its Application

In Section 2.1 we describe the basic "no arrivals" model and its analysis. Section 2.2 describes the results that we obtain through the use of this model. Variations on the basic model are considered in Section 2.3: the effect of the cost of job migration (assumed to be zero in the basic model), and the impact of assumptions made in the basic model regarding the form of the job service demand distribution.

2.1. The Basic "No Arrivals" Model

The motivation for the "no arrivals" model is our desire to determine the *maximum possible* performance benefits of migration. To understand how this model helps to achieve this goal, consider again the motivation for migration. A distribution of jobs across nodes that was balanced through initial placement will become imbalanced as shorter jobs complete and leave behind an uneven distribution of longer jobs. Migration can be used to rebalance the system by redistributing these longer jobs. Note, however, that *some* rebalancing is possible using initial placement alone: if there is a steady stream of new job arrivals, then these can be placed at the nodes that have become underloaded.

Based on the above considerations, we expect migration to offer the greatest performance benefits over non-migratory policies when (1) there is high variability in job service demands (implying that imbalancing will occur in the manner described above), and (2) the arrival process is characterized by "bursts" of job arrivals followed by long periods in which there are no job arrivals, thus preventing a load sharing policy based only on initial placement from performing any rebalancing. The second condition suggests considering a system with an extreme arrival process in which some number of jobs (a "bulk") arrives simultaneously, and then no new arrivals occur until all of these jobs have been completed. The "no arrivals model" we consider in this section allows us to study the performance of such a system by modelling the servicing of the jobs in a single bulk.

Since we wish to explore the *potential* benefits of migration, rather than the benefits afforded by any particular migration policy, we compare perfect non-migratory load sharing to perfect load sharing with migration. (In practice, the latter would be much harder to achieve than the former, so as desired we are giving the "benefit of the doubt" to migration.) Perfect non-migratory load sharing is modelled by supposing that the jobs in the bulk are (initially) evenly distributed among the nodes and that each node services its jobs in isolation (i.e., no movement of jobs from node to node occurs). (Note that this non-migratory load sharing is "perfect" only in a very restricted sense. "Truly perfect" non-migratory load sharing would use information about the expected *service demands* of resident jobs (perhaps based on accumulated service or on the "type" of job) rather than just queue length information. The "perfect" non-migratory load sharing that we use here can in fact make bad initial placement decisions, since it does not distinguish between big and small jobs.) Perfect load sharing with migration is modelled by supposing that, again, the jobs are initially evenly distributed, but also that migrations are performed to ensure that no node is allowed to become idle while two or more jobs remain at some other node. We assume that job migration can be performed at zero cost; in Section 2.3 we study the effect of the perhaps substantial cost that it would incur in reality.

Each node is represented in our model by a single-server queueing center at which jobs are scheduled using (for example) either Processor Sharing or generalized FB scheduling [Kleinrock 1976b]. In this context these scheduling disciplines give identical results. Note, however, that somewhat different results might be obtained if an (unrealistic) discipline was used in which "small" jobs are penalized to a much greater extent by the presence of "large" jobs.

Variability in job service demands is characterized in our model by the squared coefficient of variation of the service demand distribution, which we denote by CV^2. There are many distributions that match a specified mean and CV^2. We choose here a two-stage hyperexponential distribution (HE-2) consisting of two parallel exponential stages that are selected with probabilities $1-p$ and p respectively [Kleinrock 1976b]. The mean service demand of the first stage is set to zero. The mean service demand of the second stage, S, is set to $\frac{1+CV^2}{2}$. where CV^2 is the squared coefficient of variation that is to be achieved. The selection probability for the second stage, p, is fixed at $1/S$ yielding a mean service demand of one.

Our selection of zero as the service demand of the first stage requires some explanation (there are alternative choices that would still allow matching of any desired mean and CV^2). The first stage represents "small jobs", while the other stage (with mean service demand S) represents "large jobs". Intuitively, the faster the small jobs exit the system and imbalancing occurs, the larger will be the benefits of migration; thus, by choosing the service demands of the small jobs to be zero, we consider a context in which the potential benefits of migration should be maximized. The effect of the form of the service demand distribution is considered in more detail in Section 2.3.

The parameters that must then be specified in our model include the number of nodes in the system (denoted by M), the number of jobs initially allocated to each node (denoted by N; it is assumed here that the bulk size is divisible by M so that a perfectly even distribution is possible) and the squared coefficient of variation of job service demands. The mean job service demand is fixed at one. We wish to determine the impact of migration on both the *average* job residence time, and the average *maximum* job residence time (the time required to complete all the jobs in the bulk). The former measure allows us to determine the degree to which migration can

potentially improve *average performance*; the latter measure provides insight into the potential improvement in the response times of the large jobs that get completed last.

2.2. Analysis of the Model

Consider first the analysis of the model for the average job residence time in the two cases of migratory and non-migratory load sharing. Note that average job residence time is given by the total average residence time (summed over all jobs) divided by the number of jobs, and that the residence times of the small jobs are zero. We state our results in terms of N, M, and p (note that $p=\frac{2}{1+CV^2}$, and that Np gives the average number of large jobs per node).

For non-migratory load sharing, since all nodes initially have the same number N of jobs, the average job residence time over the system as a whole is identical to that at a single node, yielding

$$R_{ave}^{non-migratory} = \frac{S\sum_{k=0}^{N}\binom{N}{k}p^k(1-p)^{N-k}\sum_{j=1}^{k}j}{N}$$

(The innermost sum reflects the fact that the j-th job to complete service (among k jobs with non-zero service times) has an average residence time equal to S (the average service time of jobs with non-zero service times) multiplied by j.) Recognizing that the innermost sum is identical to $\frac{(k+1)k}{2}$ and utilizing the facts that $\sum_{k=0}^{N}\binom{N}{k}p^k(1-p)^{N-k}k = Np$ and $\sum_{k=0}^{N}\binom{N}{k}p^k(1-p)^{N-k}k^2 = Np(1-p)+(Np)^2$ results in

$$R_{ave}^{non-migratory} = \frac{S(Np(1-p)+(Np)^2+Np)}{2N}$$

Finally, noting that $S = 1/p$ and simplifying yields

$$R_{ave}^{non-migratory} = 1 + p\frac{N-1}{2} \qquad (1)$$

For load sharing with migration we have

$$R_{ave}^{migratory} = \frac{S\sum_{k=0}^{MN}\binom{MN}{k}p^k(1-p)^{MN-k}(k+\sum_{i=1}^{k-M}\frac{i}{M})}{MN}$$

(The term $k+\sum_{i=1}^{k-M}\frac{i}{M}$ reflects the fact that the total of the average job residence times is given by the total of the average service times (S multiplied by k) plus the total of the average queueing delays (S multiplied by $\sum_{i=1}^{k-M}\frac{i}{M}$). The expression for queueing delays is perhaps most easily understood by considering the case of FCFS scheduling (which, once the jobs with zero service times have completed, yields average job residence times identical to those of Processor Sharing or generalized FB).) Recognizing that the innermost sum is identical to $\frac{(k-M)(k-M+1)}{2M}$ for $k \geq M$ and utilizing the facts that $\sum_{k=0}^{MN}\binom{MN}{k}p^k(1-p)^{MN-k} = 1$, $\sum_{k=0}^{MN}\binom{MN}{k}p^k(1-p)^{MN-k}k = MNp$, and $\sum_{k=0}^{MN}\binom{MN}{k}p^k(1-p)^{MN-k}k^2 = MNp(1-p)+(MNp)^2$ results in

$$R_{ave}^{migratory} = \frac{S}{MN}\times\left[\frac{M(M-1)}{2M}+(1-\frac{(2M-1)}{2M})MNp+\frac{MNp(1-p)+(MNp)^2}{2M} - \sum_{k=0}^{M-1}\binom{MN}{k}p^k(1-p)^{MN-k}\frac{(k-M)(k-M+1)}{2M}\right]$$

Noting that $S = 1/p$ and simplifying yields

$$R_{ave}^{migratory} = \frac{1}{2}\left[Np+\frac{1}{Np}\right]+\frac{1-\frac{p}{2}-\frac{1}{2Np}}{M} - \frac{1}{MNp}\sum_{k=0}^{M-1}\binom{MN}{k}p^k(1-p)^{MN-k}\frac{(k-M)(k-M+1)}{2M} \qquad (2)$$

Consider next the analysis of the model for the average maximum residence time. For load sharing with migration, noting that the maximum of k exponentials of mean one has mean value $\sum_{l=1}^{k}\frac{1}{l}$, we have

$$R_{max}^{migratory} = S\left[\sum_{k=0}^{M-1}\binom{MN}{k}p^k(1-p)^{MN-k}\sum_{l=1}^{k}\frac{1}{l} + \sum_{k=M}^{MN}\binom{MN}{k}p^k(1-p)^{MN-k}(\frac{k-M}{M}+\sum_{l=1}^{M}\frac{1}{l})\right]$$

which can be simplified to

$$R_{max}^{migratory} = N + \frac{1}{p}\left[\sum_{l=2}^{M}\frac{1}{l} - \sum_{k=0}^{M-1}\binom{MN}{k}p^k(1-p)^{MN-k}(\frac{k-M}{M}+\sum_{l=k+1}^{M}\frac{1}{l})\right] \qquad (3)$$

For non-migratory load sharing an exact analysis is not possible. The values used in the next section were obtained through simulation by taking midpoints of 98% confidence intervals of width (at most) plus or minus 4% (obtained using the method of independent replications).

2.3. Results

Using the basic "no arrivals" model just described, one can assess the value of migration by the percentage improvements in average residence times and (average) maximum residence times that load sharing with migration yields in comparison to non-migratory load sharing. These percentage improvements are shown in Figures 1 and 2 in the form of *contour diagrams*. Each figure contains one contour diagram for 3, 10, 30, and 100 node systems. The x and y axes of each contour diagram give the squared coefficient of variation of job service demands and the number of jobs initially present at each node, respectively. Each contour line joins together points of equal percentage improvement; we have chosen here to draw contour lines at 5% spacings.

Note that only CV^2 values of greater than or equal to one can be attained by the HE-2 distribution used in our model; thus, the effective range on the x axis of each contour diagram starts from one. An upper limit of 30 for CV^2 has been used in the contour diagrams for average residence times. This limit was reduced to 15 in the contour diagrams for maximum residence times because of the need to use simulation in generating these diagrams, and the extremely long run lengths high CV^2 values imply. Even with this reduced range, and long run lengths, statistical fluctuations still cause some raggedness in the contour lines, particularly in those areas when the percentage improvement is changing very slowly.

All of the contour diagrams display a similar structure, in which for increasing CV^2 and fixed N, as well as for increasing N and fixed CV^2, the percentage improvement initially increases, attains a maximum, and then starts to decrease. The contour diagrams for maximum residence time differ somewhat from those for average residence time in that the decrease observed with increasing N and fixed CV^2 (beyond the point at which the percentage improvement in maximized) is much more gradual.

This common structure can be explained as follows. First, consider a fixed N. For CV^2 close to one, almost all jobs are large jobs, and thus the system is still almost perfectly balanced after the small jobs complete. As CV^2 increases, so does the proportion of jobs that are small. (This is true regardless of the form of the distribution that is employed, assuming that a fixed mean is maintained.) Therefore, as CV^2 increases, so does the imbalance in the system that results from the completion of the small jobs, and thus so does the benefit of migration. However, for sufficiently large CV^2 it becomes unlikely that a single node would have more than one large job, and the benefits of migration again become small.

Now, consider a fixed CV^2. For small N, it would be unlikely that a single node would have more than one large job; thus, little imbalancing occurs and migration provides little benefit. As N increases, the likelihood of imbalancing increases, and thus so does the benefit of migration. Note, however, that migration only influences "end-effects" – it provides benefits only after at least one

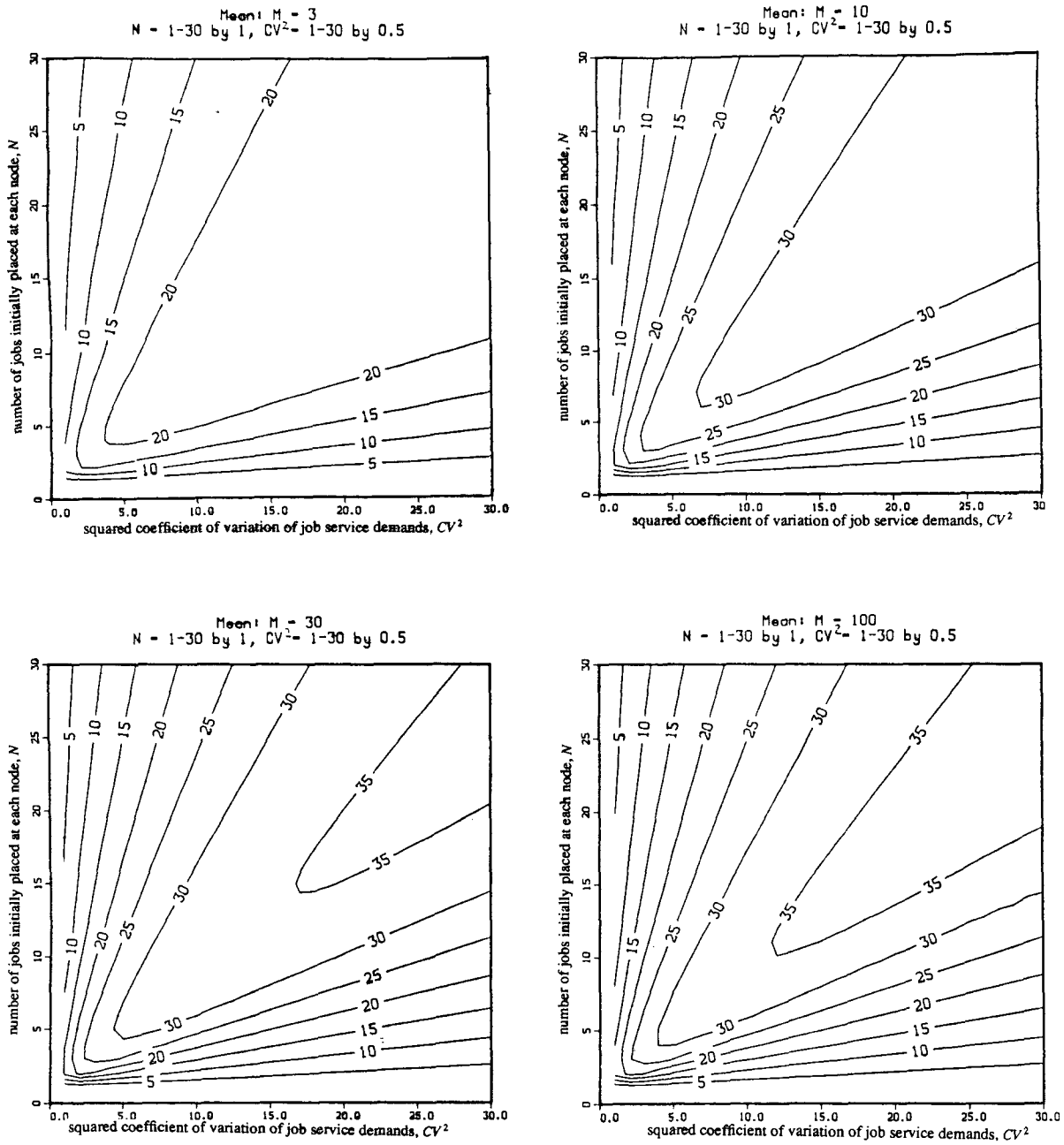


Figure 1 (a)–(d). Contours of Percentage Improvement in Average Residence Time for Migratory Load Sharing Over Non-Migratory Load Sharing (3, 10, 30, and 100 nodes).

node's queue has emptied. Thus, for sufficiently large N the benefits of migration again become small, since the number of jobs that are affected by these end-effects, although larger in an absolute sense, becomes an insignificant portion of the total.

Figures 1 and 2 allow us to make the following observations regarding the potential benefits of migration as indicated by our model:

- The potential benefits of migration with respect to average and maximum residence times are surprisingly similar. (One would perhaps have expected migration to offer much greater improvements in maximum residence times.)
- The potential benefits of migration seem bounded by relatively small values (no percentage improvement over 40% was encountered). The causes for these limits are partially exposed through the common structure of the contour graphs (in particular, the indicated behavior as CV^2 is fixed and N is varied) – migration influences only "end-effects" that occur when at least one node has emptied its queue, yet such end-effects tend to be the most pronounced (affect the largest *number* of jobs) precisely when they have the least effect (affect the smallest *proportion* of jobs).

Although the *maximum* potential benefit of migration is an increasing function of CV^2, it increases only gradually beyond a CV^2 of around 5. (For example, from Figure 1 the maximum percentage improvement in average response time with 30 nodes and a CV^2 of 5 is not much less than that with 30 nodes and a CV^2 of 35.) This observation provides a different way to view the

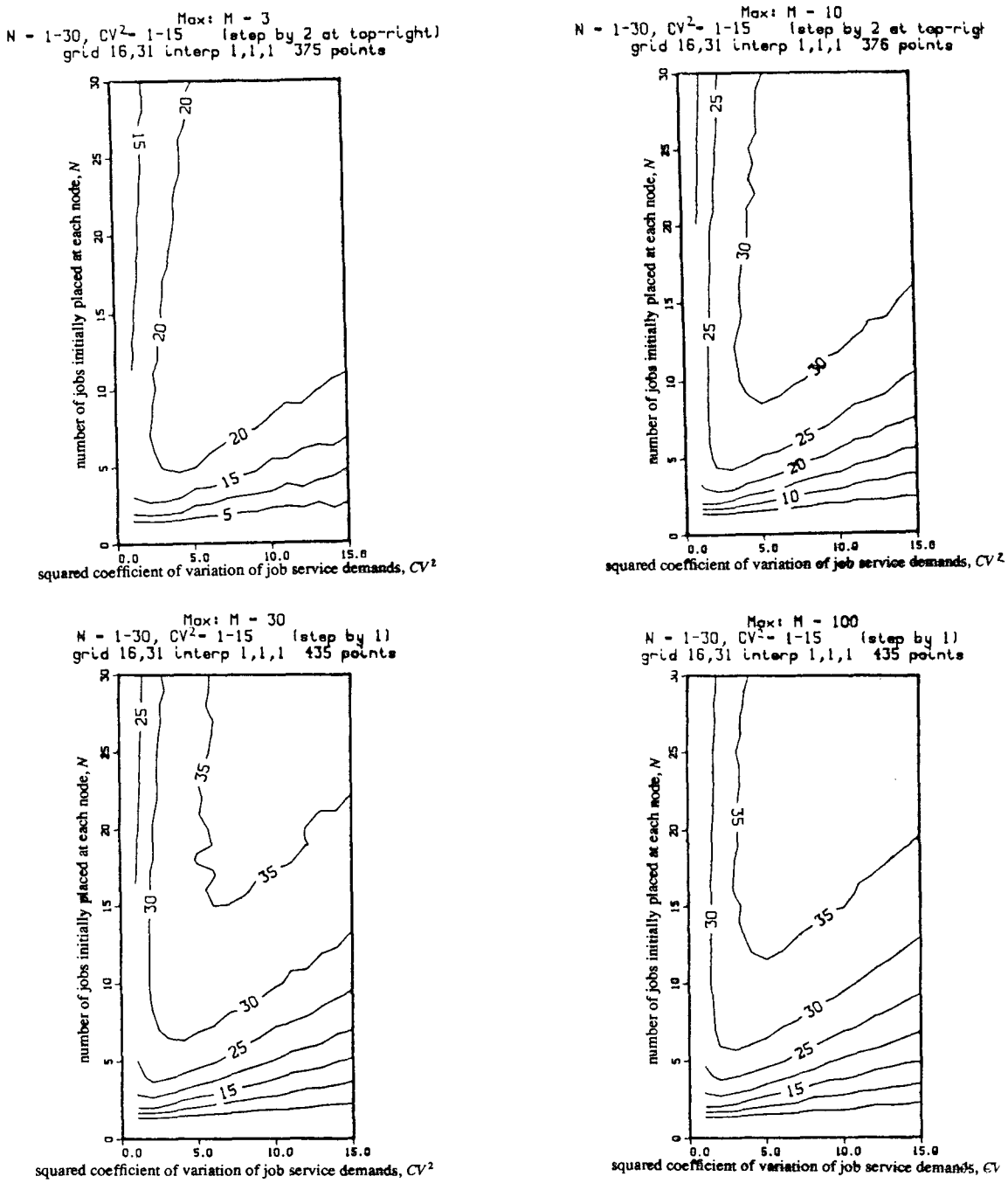


Figure 2 (a)–(d). Contours of Percentage Improvement in Maximum Residence Time (3, 10, 30, and 100 nodes).

limited potential benefit of migration: workload variability increases this potential benefit, but in a way that is strongly subject to the law of diminishing returns.

The second of these observations can be strengthened somewhat by bounding analytically the percentage improvement in average residence time provided by migration (within our model). It is straightforward to show that this percentage improvement is an increasing function of the number of nodes M. Thus, consider equation (2) for $M \to \infty$. The Laplace approximation to the binomial probability mass function [Trivedi 1982] is asymptotically exact in this case, and we have, therefore, for $M \to \infty$

$$R_{ave}^{migratory} \to \frac{1}{2}\left[Np + \frac{1}{Np}\right] + \frac{1 - \frac{p}{2} - \frac{1}{2Np}}{M} - \frac{1}{MNp}\sum_{k=0}^{M-1} \frac{1}{\sqrt{2\pi MNp(1-p)}} e^{\frac{-(k-MNp)^2}{2MNp(1-p)}} \frac{(k-M)(k-M+1)}{2M}$$

For $Np > 1$, each term in the sum goes to zero faster than $\frac{1}{M}$ as $M \to \infty$. Since there are only M of these terms, as $M \to \infty$ their sum goes to zero as well. Thus, as $M \to \infty$ we have, for $Np > 1$,

$$R_{ave}^{migratory} \to \frac{1}{2}\left(Np + \frac{1}{Np}\right)$$

Since, for fixed p, $R_{ave}^{migratory}$ for $Np<1$ can be no greater than $R_{ave}^{migratory}$ for $Np=1$, the above expression shows that $R_{ave}^{migratory}$ for $Np<1$ can have a limit no greater than 1 as $M\to\infty$. But $R_{ave}^{migratory}$ must be at least 1. Thus, for $Np<1$, $R_{ave}^{migratory}$ must tend to 1 as $M\to\infty$. This yields, for $M\to\infty$ and all Np,

$$R_{ave}^{migratory} \to \frac{1}{2}(\max(Np,1) + \frac{1}{\max(Np,1)}) \quad (4)$$

Therefore, the percentage improvement in average residence time tends to (using equation (1))

$$\frac{1+p\frac{N-1}{2} - \frac{1}{2}(\max(Np,1) + \frac{1}{\max(Np,1)})}{1+p\frac{N-1}{2}}$$

It is straightforward to show that, for fixed p, this expression is maximized by the following (in general non-integral) value of N

$$N = \frac{1+\sqrt{5-4p+p^2}}{2p-p^2} \quad (5)$$

Noting that the percentage improvement in average residence time given above is maximized as $p\to 0$, this last expression allows us to obtain the value of Np that maximizes this percentage improvement (equal to $\frac{1+\sqrt{5}}{2}$), and allows us to conclude that the percentage improvement in average residence time is bounded above by $\frac{3-\sqrt{5}}{2}$, a quantity somewhat less than 40% (and in good agreement with the results shown in Figure 1).

2.4. Variations on the Basic Model

The basic "no arrivals" model relied on a number of assumptions regarding the cost of job migration and the form of the job service demand distribution. In this section we consider the influence of these factors and the consequences of making alternative assumptions about them.

2.4.1. The Effect of Job Migration Cost

The cost of job migration is assumed to be zero in the basic model. In practice, however, this cost could be substantial. Here we consider incorporating this cost in our model. To achieve analytical tractability, we account only for the cost of those migrations that would be required to rebalance the system when the small jobs depart (and the initial system balance is destroyed). We do not account for any job migrations performed to correct imbalances resulting from variability in the service times of the large jobs. For example, the total migration cost that we will obtain for a CV^2 of one, for which there are no small jobs, is zero. However, we expect this approach to be increasingly accurate as CV^2 is increased, and, in any case, it illustrates well the effect of job migration costs on the primary justification for migration – the desire to perform system rebalancing.

The cost of performing a job migration is reflected as a processor cost C at the source node. (Other assumptions, such as an identical cost at both source and destination nodes, are possible but do not change the nature of the results.) Assuming that migrations are performed immediately after the small jobs (instantaneously) complete, and that the number of nodes M in the system is very large, the probability that a given node must perform j migrations ($j\geq 1$) is given by $\binom{N}{k}p^k(1-p)^{N-k}$ where $k=\lceil Np\rceil+j$. (Note that in the balanced state resulting after migrations are performed, all but a vanishingly small proportion of nodes will have either $\lfloor Np\rfloor$ or $\lceil Np\rceil$ large jobs.) The average (over all MN jobs) additional delay experienced due to job migrations (including the queueing time caused by other jobs being migrated) is then given by

$$C\sum_{k=\lceil Np\rceil+1}^{N}\binom{N}{k}p^k(1-p)^{N-k}\sum_{j=1}^{k-\lceil Np\rceil}(k-(j-1)) \quad (6)$$

The contour diagram in Figure 3 shows contours (at 10% spacings) of equal percentage improvement in average residence time, for large M (see expression (4)), when the delay cost of job migration given by expression (6) has been included in the residence time for load sharing with migration. For each value of CV^2 (x axis), the value of N used is the one that maximizes the benefit of migration (see equation (5)). On the y axis is given the processor cost C of a single job migration. (The peculiar structure that occurs at a CV^2 of one is due to the fact that no initial rebalancing, and thus no migrations whose cost we account for, can occur here.)

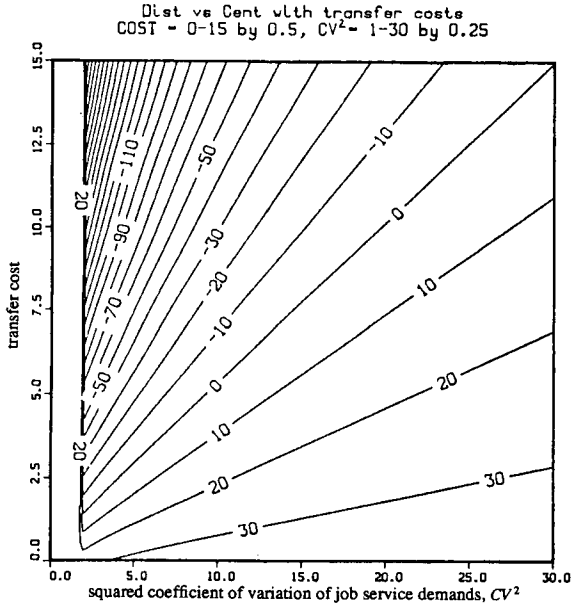


Figure 3. Contours of Percentage Improvement in Average Residence Time When Transfer Cost is Included.

The major observation regarding this figure is the limited influence of the job migration cost on the performance benefits of migration. Noting that the average job service demand is fixed at one, observe the scale on the y axis. For that range of CV^2 in which migration offers its greatest benefits (e.g., around a CV^2 of 10), the job migration cost must significantly exceed the average job service demand in order to significantly limit the performance benefits of migration. Intuitively, this can be explained by the fact that only large jobs are migrated, and that the job migration cost can therefore be substantial and yet still be minor compared to the service demands of the migrating jobs (and thus of limited effect). Note that the 0% benefit line falls almost exactly where the processor cost C of a job migration equals 50% of the average service demand of the large jobs that may be migrated (as given by S, which equals $\frac{1+CV^2}{2}$).

In conclusion, it seems reasonable (perhaps surprisingly) to assess the potential benefits of migration using a model that neglects job migration cost, although this requires some faith in the ability of real policies to in fact identify the large jobs that are the good candidates for migration. We now consider the impact of other assumptions made in the basic "no arrivals" model; specifically, those regarding the form of the job service demand distribution.

2.4.2. The Effect of the Job Service Demand Distribution: Exponential Stages

Recall that the job service demand distribution that we use in our basic model is an HE-2 distribution with one of the two parallel, exponential stages having zero mean service demand. There are two questions we consider regarding the form of this distribution. First, what is the impact of allowing no "intermediate-sized" jobs? (With this distribution, either jobs have service demand zero, or have a demand chosen from an exponential distribution with a large mean.) Second, what is the impact of the exponential distribution used for

each stage? (In particular, does the long tail of the exponential distribution greatly influence our results?)

We consider the latter of these questions first, through a discrete service demand distribution in which jobs have either a demand of zero or a fixed demand S. As usual, we fix the mean demand at one; hence, the probability that a job has demand S (denoted by p) is given by $\frac{1}{S}$. S is fixed so as to yield a given CV^2 using the equation $S = CV^2+1$.

It is not clear what scheduling strategy is most appropriately assumed, given the above job service demand distribution. Again, Processor Sharing and generalized FB scheduling yield the same results. However, these disciplines must schedule (after the small jobs depart) a collection of identical jobs, a context for which they (unnaturally) assume the worst possible scheduling strategy. An alternative that perhaps yields more realistic performance is First-Come-First-Served (FCFS) with preemptive priority given to the small jobs. This is the discipline that we assume here. (Note that for the job service demand distribution used in the basic model, this discipline would yield the same results as those that we derived for this model; i.e., identical results as with Processor Sharing or generalized FB.) Finally, we assume, in the case of load sharing with migration, that rebalancing is performed immediately after the (instantaneous) departure of the small jobs.

With these assumptions, it is straightforward to show that

$$R_{ave}^{nonmigratory} = \frac{S\sum_{k=0}^{N}\binom{N}{k}p^k(1-p)^{N-k}\sum_{j=1}^{k}j}{N}$$

and

$$R_{ave}^{migratory} = \frac{S\sum_{k=0}^{MN}\binom{MN}{k}p^k(1-p)^{MN-k}\left(M\sum_{j=1}^{\left\lfloor\frac{k}{M}\right\rfloor}j + (k-M\left\lfloor\frac{k}{M}\right\rfloor)(1+\left\lfloor\frac{k}{M}\right\rfloor)\right)}{NM}$$

The same maximum residence times are obtained for any work conserving scheduling discipline, and can be derived as

$$R_{max}^{nonmigratory} = S\sum_{k=0}^{N}k\left[\left[\sum_{j=0}^{k}\binom{N}{j}p^j(1-p)^{N-k}\right]^M - \left[\sum_{j=0}^{k-1}\binom{N}{j}p^j(1-p)^{N-k}\right]^M\right]$$

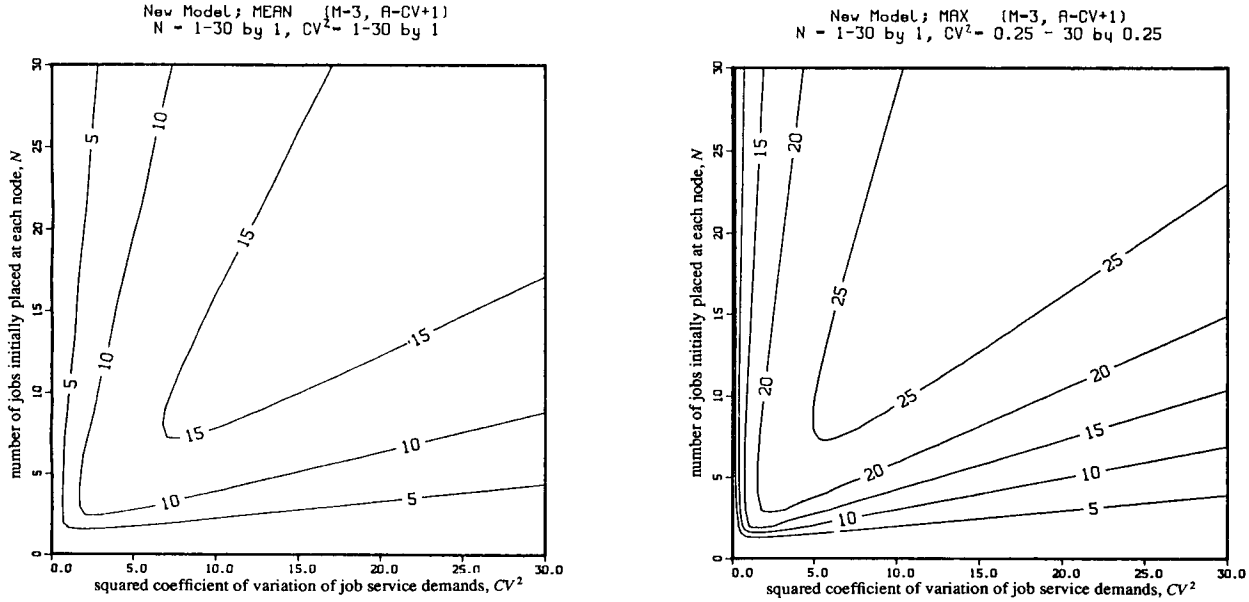
and

$$R_{max}^{migratory} = S\sum_{k=0}^{MN}\binom{MN}{k}p^k(1-p)^{MN-k}\left\lfloor\frac{k}{M}\right\rfloor$$

Figures 4 and 5 display contour diagrams for the percentage improvements in average residence time and maximum residence time, respectively, for a 3 node system, using the alternative job service demand distribution described above. A comparison of the contour diagrams of Figures 4 and 5 to those for the job service demand distribution used in the basic model, shown in Figure 1 (a) and Figure 2 (a), suggests that, in fact, the use of the exponential distribution for each stage of the HE-2 had little impact on our results. Somewhat lesser improvements are exhibited for average residence time in Figure 4 than in Figure 1 (a), while somewhat greater improvements are exhibited in Figure 5 for maximum residence time than in Figure 2 (a). However, the form of the results is very similar.

2.4.3. The Effect of the Job Service Demand Distribution: Absence of "Intermediate-Sized" Jobs

We now turn to the other question we identified regarding the form of the job service demand distribution – what is the impact of allowing no "intermediate-sized" jobs? This issue is studied through a discrete job service demand distribution similar to that used in Section 2.4.2, but in which jobs may now have a demand identical to the mean service demand of one, rather than just a demand of zero or S. As before, the probability that a job has demand S is denoted by p. The probability that a job has demand one is denoted by q. The additional parameters in the distribution allow a fixed choice of S that will serve for all CV^2 values that will be considered; this value is chosen here to be 32, the smallest value that will allow all CV^2 values in the range 0 to 30 to be covered while still ensuring a non-zero probability of a job service demand of one. To achieve a mean demand of one and a given CV^2 requires that q and p be chosen so that $q = 1 - \frac{CV^2}{S-1}$ and $p = \frac{1-q}{S}$. In our numerical experiments values of N no greater than 30 were utilized, implying that the total service demand of all intermediate-sized jobs at a node can never exceed the demand of a single large job. The expression given below for $R_{max}^{nonmigratory}$ depends on this fact.



Figures 4 & 5. Contours of Percentage Improvement in Average and Maximum Residence Times for an Alternative Service Demand Distribution (cf. Figures 1 (a) and 2 (a), respectively).

Again, we assume FCFS with preemptive priority. In this case, the jobs with service demand zero have preemptive priority over all other jobs, while the jobs with service demand one have preemptive priority over the large jobs with service demand S. We require a somewhat more detailed assumption about how migration is performed; specifically, whether intermediate-sized jobs are migrated. We assume the most favorable possible scenario for migration, in which system balancing is performed both with respect to the intermediate-sized jobs, and with respect to the large jobs. Again, we assume that this balancing is performed immediately after the departure of the jobs with zero service times. With these assumptions, it is straightforward to show that

$$R_{ave}^{nonmigratory} = \frac{\sum_{k=0}^{N}\sum_{j=0}^{N-k}\binom{N}{k}\binom{N-k}{j}(1-p-q)^{N-k-j}p^k q^j \left[\sum_{l=1}^{j} l + kj + S\sum_{l=1}^{k} l\right]}{N}$$

and

$$R_{ave}^{migratory} = \frac{1}{NM}\sum_{k=0}^{MN}\sum_{j=0}^{MN-k}\binom{MN}{k}\binom{MN-k}{j}(1-p-q)^{MN-k-j}p^k q^j \times$$

$$\left(M\sum_{l=1}^{\left\lfloor \frac{j}{M} \right\rfloor} l + k\left\lfloor \frac{j}{M} \right\rfloor + (j - M\left\lfloor \frac{j}{M} \right\rfloor)(1+\left\lfloor \frac{j}{M} \right\rfloor + \left\lfloor \frac{k}{M} \right\rfloor) \right.$$

$$+ MS\sum_{l=1}^{\left\lfloor \frac{k}{M} \right\rfloor} l + S(k - M\left\lfloor \frac{k}{M} \right\rfloor)(1+\left\lfloor \frac{k}{M} \right\rfloor)$$

$$\left. + \max(0, k - M\left\lfloor \frac{k}{M} \right\rfloor - (M - (j - M\left\lfloor \frac{j}{M} \right\rfloor))) \right)$$

Identical maximum residence times are obtained for any work conserving scheduling discipline, and can be derived as

$$R_{max}^{nonmigratory} = \sum_{k=0}^{N}\sum_{j=0}^{N-k}(kS+j) \times$$

$$\left(\left[\binom{N}{k}p^k \sum_{i=0}^{j}\binom{N-k}{i}(1-p-q)^{N-k-i}q^i + \sum_{i=0}^{k-1}\binom{N}{i}p^i(1-p)^{N-i} \right]^M - \right.$$

$$\left. \left[\binom{N}{k}p^k \sum_{i=0}^{j-1}\binom{N-k}{i}(1-p-q)^{N-k-i}q^i + \sum_{i=0}^{k-1}\binom{N}{i}p^i(1-p)^{N-i} \right]^M \right)$$

and

$$R_{max}^{migratory} = \sum_{k=0}^{MN}\sum_{j=0}^{MN-k}\binom{MN}{k}\binom{MN-k}{j}(1-p-q)^{MN-k-j}p^k q^j \times$$

$$\left(\left\lceil \frac{j}{M} \right\rceil + S\left\lceil \frac{k}{M} \right\rceil - \alpha_{0<k-M\left\lfloor \frac{k}{M} \right\rfloor <M-(j-M\left\lfloor \frac{j}{M} \right\rfloor)} \right)$$

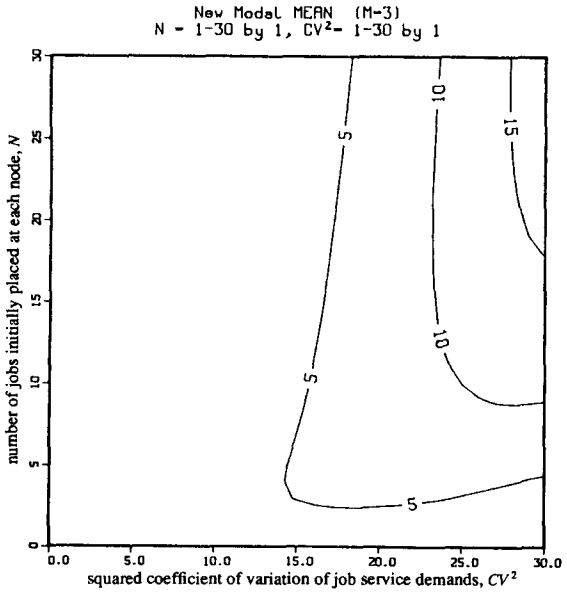
where $\alpha_{0<k-M\left\lfloor \frac{k}{M} \right\rfloor <M-(j-M\left\lfloor \frac{j}{M} \right\rfloor)}$ equals one if the condition defined in the subscript holds and zero otherwise.

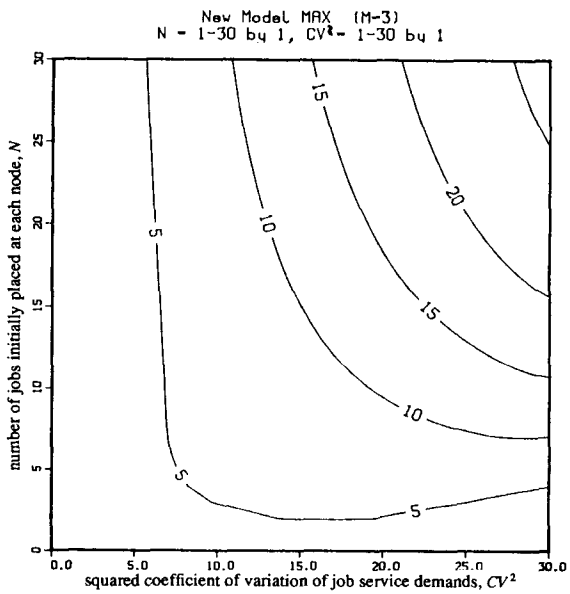
Figures 6 and 7 display contour diagrams for the percentage improvements in average residence time and maximum residence time, respectively, for a 3 node system. A comparison of the contour diagrams of Figures 6 and 7 to those for the job service demand distribution used in the basic model, shown in Figure 1 (a) and Figure 2 (a), suggests that, in fact, the presence or absence of intermediate-sized jobs *is* an important workload characteristic. With such jobs present, the potential benefits of migration are reduced considerably. In addition, the form of our results changes in a significant manner – the performance benefits of migration no longer peak at relatively small CV^2 as in Figures 1 and 2. Intuitively, migration is less useful in this context since, for a given CV^2, there is a smaller proportion of large jobs (the jobs migration is most usefully applied to), and thus these jobs are less important in the sense of contributing to average and maximum residence times.

3. An Open Queueing Model and its Application

The "no arrivals" model described in the previous section yielded insight regarding the potential performance benefits of migration, and the dependence of these benefits on workload characteristics such as the coefficient of variation of job service demands. However, it provided little information about how the potential benefits of migration depend on the system's job arrival process. Also, while it did provide a comparison of migratory load sharing to non-migratory load sharing, it did not place this comparison in perspective by comparing the two approaches to load sharing to the case of no load sharing.

The open queueing model utilized here allows us to address these issues. As before, each node is represented by a single-server queueing center, at which jobs are scheduled using (for example)





Figures 6 & 7. Contours of Percentage Improvement in Average and Maximum Residence Times with Intermediate-Sized Jobs (cf. Figures 1 (a) and 2 (a), respectively).

either Processor Sharing or generalized FB scheduling. The same job service demand distribution that was utilized in the basic "no arrivals" model is utilized here. Now, however, we assume an arrival process of jobs at each node, with some specified mean and squared coefficient of variation of interarrival times. The specific interarrival time distribution used is a two-stage hyperexponential distribution of the same form as that used for job service demands. This yields a bulk-type arrival process, thus providing optimistic estimates of the benefits of migration.

Simulation was employed to obtain average residence time estimates for optimal, costless load sharing with migration and optimal, costless non-migratory load sharing. (More specifically, "optimal" assuming no knowledge of job service demands.) An optimal policy for load sharing with migration (with our job service demand distribution) is simply to instantaneously migrate a job to any node that becomes idle, if there are two or more jobs at some other node. An optimal policy for non-migratory load sharing is to route each new arrival to the node with the shortest queue (being careful not to discard the jobs with zero service times until all the jobs in a particular bulk of arrivals have been assigned to nodes). Since no costs are incorporated, our results will again tend to be optimistic for load sharing with migration.

Performance results were obtained analytically for the situation in which no load sharing takes place. In this case, each node can be analyzed independently as a GI/GI/1 queue. Although difficult to analyze in general, the special forms of the arrival and service distributions that we use allow a simple closed form solution. Note first that it is sufficient to solve for the average residence time of the large jobs, since the small jobs have a known average residence time (zero). Also, the average residence time of the large jobs is identical to that in a system in which the workload consists of large jobs only (and the arrival rate of jobs has been correspondingly reduced). Thus, we need only solve a G/M/1 model with a reduced arrival rate. The simple form of the interarrival time distribution allows us to solve this model using standard transform methods [Kleinrock 1976a], yielding, as the solution of our original queueing model

$$R_{ave}^{noloadsharing} = \frac{CV^2_s + CV^2_a}{(1 + CV^2_s)(1-\rho)}$$

where ρ denotes the mean service demand multiplied by the mean arrival rate, CV^2_s denotes the squared coefficient of variation of the job service demand, and CV^2_a denotes the squared coefficient of variation of the job interarrival time. (Both coefficients of variation are constrained to be greater than or equal to one.) It may seem anomalous that average residence time is a decreasing function of CV^2_s for CV^2_a greater than one – but recall that the scheduling discipline we assume is one that *benefits* from increased variability in service demands.

In Table 1 we show sample average residence time results for various values of CV^2_a and CV^2_s, for a system with 10 nodes, a mean service demand of one, and an arrival rate of .75. Observe that the percentage improvement in average residence time provided by load sharing with migration, in comparison to non-migratory load sharing, is monotonically increasing in CV^2_s, for a given CV^2_a (allowing for statistical fluctuations). Similarly, the percentage improvement is monotonically increasing in CV^2_a, for a given CV^2_s. However, note that these percentage improvements are, in general, quite small in comparison to the percentage improvement non-migratory load sharing offers in comparison to the absence of load sharing. Also, noting that high CV^2_s and CV^2_a are required before significant improvements are observed, and recalling that a number of assumptions were made that are optimistic for migration (for example, regarding the forms of the interarrival time and service demand distributions), these results would seem to indicate that migration has little to offer as far as improving average system performance in practice.

CV^2_a	CV^2_s	Load Sharing with Migration	Non-Migratory Load Sharing	No Load Sharing
1	1	1.12	1.24	4.00
3	1	1.40	1.61	8.00
3	3	1.25	1.48	6.00
3	10	1.17	1.38	4.73
3	30	1.16	1.36	4.26
3	100	1.15	1.36	4.08
10	1	2.66	2.99	22.00
10	3	1.83	2.21	13.00
10	10	1.36	1.80	7.27
10	30	1.20	1.66	5.16
10	100	1.13	1.63	4.36
30	1	5.93	6.58	62.00
30	3	3.61	4.09	33.00
30	10	2.04	2.70	14.55
30	30	1.43	2.12	7.74
30	100	1.27	1.95	5.15

Table 1. Average Residence Times in an Open Queueing Model.

4. Conclusions

We have used a combination of analysis and simulation to examine the performance benefits offered by migratory load sharing beyond those offered by non-migratory load sharing. This question is interesting because migratory load sharing seems advantageous intuitively, but the facilities to support it efficiently are not present in many operating systems. We conclude that:

- There are likely *no* conditions under which migration could yield *major* performance improvements beyond those offered by non-migratory load sharing, particularly when viewed relative to the advantages of non-migratory load sharing over no loadsharing.
- Under some fairly extreme conditions, migration *can* offer *modest* additional performance improvements. These extreme conditions are characterized by high variability in *both* job service demands and the workload generation process.
- The benefits of migration are not limited by its cost, but rather by the inherent effectiveness of non-migratory load sharing. Whenever migration does offer significant potential benefits, the migration cost would likely be dwarfed by the (very large) service demands of the jobs that can be fruitfully migrated.
- Different job service demand distributions that match with respect to both mean and variance may yield quite different results concerning the benefits of migration, thus caution is needed when developing workload models for use in migration studies.

Our results are consistent with those of Leland and Ott [1986], whose trace-driven simulations of migration show a modest improvement in the performance of large jobs and no significant improvement in overall average performance (cf. their Figure 9). However, the results of our analytic study encompass much broader dimensions of workload characteristics than are embodied in the traces used by Leland and Ott.

We acknowledge that the ability to migrate active processes may be important for reasons other than load sharing performance – for example, to allow long-running processes to be moved from a machine that requires maintenance, or to allow a returning workstation owner to banish freeloaders. Recognizing the limited performance benefits of migration is important, though, particularly because this means that "costlier but simpler" implementations of migration may be acceptable.

Acknowledgements

We discussed this subject extensively with Brigitte Plateau. Darryl Willick assisted us in obtaining our numerical results.

References

[Barak & Litman 1984]
A. Barak and A. Litman. MOS: A Multicomputer Distributed Operating System. Department of Computer Science, The Hebrew University of Jerusalem, 1984.

[Barak & Shiloh 1985]
A. Barak and A. Shiloh. A Distributed Load Balancing Policy for a Multicomputer. *Software – Practice and Experience 15*,9 (September 1985), pp. 901-913.

[Bershad 1985]
B. Bershad. Load Balancing with Maitre d'. Report UCB/CSD 85/276, Computer Science Division, University of California at Berkeley, 1985.

[Bryant & Finkel 1981]
R. Bryant and R. Finkel. A Stable Distributed Scheduling Algorithm. *Proc. 2nd International Conference on Distributed Computing Systems* (April 1981), pp. 314-323.

[Eager, Lazowska & Zahorjan 1986a]
D. Eager, E. Lazowska and J. Zahorjan. Adaptive Load Sharing in Homogeneous Distributed Systems. *IEEE Transactions on Software Engineering SE-12*,5 (May 1986), pp. 662-675.

[Eager, Lazowska & Zahorjan 1986b]
D. Eager, E. Lazowska and J. Zahorjan. A Comparison of Receiver-Initiated and Sender-Initiated Adaptive Load Sharing. *Performance Evaluation 6*,1 (March 1986), pp. 53-68.

[Hagmann 1986]
R. Hagmann. Process Server: Sharing Distributed Power in a Workstation Environment. *Proc. 6th International Conference on Distributed Computing Systems* (May 1986), pp. 260-267.

[Hsu & Liu 1986]
C-Y. Hsu and J. Liu. Dynamic Load Balancing Algorithms in Homogeneous Distributed Systems. *Proc. 6th International Conference on Distributed Computing Systems* (May 1986), pp. 216-223.

[Hwang et al. 1982]
H. Hwang, W. Croft, G. Goble, B. Wah, F. Briggs, W. Simmons and C. Coates. A UNIX-Based Local Computer Network with Load Balancing. *IEEE Computer 15*,4 (April 1982), pp. 55-66.

[Kleinrock 1976a]
L. Kleinrock. *Queueing Systems: Volume 1, Theory*. John Wiley & Sons, 1976.

[Kleinrock 1976b]
L. Kleinrock. *Queueing Systems: Volume 2, Computer Applications*. John Wiley & Sons, 1976.

[Krueger & Finkel 1984]
P. Krueger and R. Finkel. An Adaptive Load Balancing Algorithm for a Multicomputer. Technical Report 539, Computer Science Department, University of Wisconsin – Madison, 1984.

[Kruskal & Weiss 1985]
C. Kruskal and A. Weiss. Allocating Independent Subtasks on Parallel Processors. *IEEE Transactions on Software Engineering SE-11*,10 (October 1985), pp. 1001-1016.

[Lee & Towsley 1986]
K. Lee and D. Towsley. A Comparison of Priority-Based Decentralized Load Balancing Policies. *Proc. Performance '86 and ACM SIGMETRICS 1986* (May 1986), pp. 70-77.

[Leland & Ott 1986]
W. Leland and T. Ott. Load-Balancing Heuristics and Process Behavior. *Proc. Performance '86 and ACM SIGMETRICS 1986* (May 1986), pp. 54-69.

[Litzkow 1987]
M. Litzkow. Remote Unix: Turning Idle Workstations Into Cycle Servers. *Proc. Summer 1987 Usenix Conference* (June 1987), pp. 381-384.

[Livny & Melman 1982]
M. Livny and M. Melman. Load Balancing in Homogeneous Broadcast Distributed Systems. *Proc. ACM Computer Network Performance Symposium* (April 1982), pp. 47-55.

[Nichols 1987]
D. Nichols. Using Idle Workstations in a Shared Computing Environment. *Proc. 11th ACM Symposium on Operating Systems Principles* (November 1987).

[Powell & Miller 1983]
M. Powell and B. Miller. Process Migration in DEMOS/MP. *Proc. 9th ACM Symposium on Operating Systems Principles* (October 1983), pp. 110-119.

[Theimer, Lantz & Cheriton 1985]
M. Theimer, K. Lantz and D. Cheriton. Preemptable Remote Execution Facilities for the V-System. *Proc. 10th ACM Symposium on Operating Systems Principles* (December 1985), pp. 2-12.

[Trivedi 1982]
K.S. Trivedi. *Probability & Statistics with Reliability, Queueing, and Computer Science Applications*. Prentice Hall, 1982.

[Wang and Morris 1985]
Y-T. Wang and R. Morris. Load Sharing in Distributed Systems. *IEEE Transactions on Computers C-34*,3 (March 1985), pp. 204-217.