ON OPTIMAL FILE ALLOCATION WITH SHARING

S. Tripathi, Y. Huang and D. Towsley

COINS Technical Report 88-93

On Optimal File Allocation with Sharing

Satish K. Tripathi *

Yennun Huang *

and

Don Towsley †

Abstract

In allocating files of a workstation (user) to file servers in a network, there exist optimal policies that place all files for a given user on a single file server [19]. Such policies, however, are only optimal if no file can be shared by more than one user. In most realistic cases, however, file sharing is common. In this paper, we show that a modified *vertex allocation theorem* provides optimal performance when shared files are preallocated. The paper presents heuristics to determine the workload for each of the file servers. The behavior of the heuristics is studied through numerous examples.

1 Introduction:

Local Area Networks (LAN's) are one of the major recent developments in computer systems. They allow resources such as print servers and file servers to be shared by

*Dept. of Computer Science, University of Maryland, College Park, MD 20742, tripathi@mimsy.umd.edu, (301)454-5165

†Dept. of Computer & Information Science, University of Massachusetts, Amherst, MA 01003.
Prof. Towsley's work was supported in part by a gift from Digital Equipment Corporation

many workstations. In this paper, we consider the allocation of workstations to file servers so as to maximize the total utilization of these file servers.

Workstations access both private files and files that are shared among many workstations. We consider the problem of allocating *private files* of various workstations among file servers so as to maximize system throughput. We assume that the shared files have been preallocated. In order to solve the allocation problem we model the combined workstation file server system as a multiple class queueing network with *only* one customer (workstation) in each class. The model assumes that the communication medium is lightly loaded and ignores the queueing effects in the communication medium.

Optimal allocation in queueing networks has been studied by several authors [3], [16], [10], [12], [13], [4], [18], [19], [14]. The routing probabilities that give the minimum delay for an open network have been studied in [7]. Whittle [18] has characterized routing probabilities which maximize the input rate that saturates an open network. Trivedi, *et al.* [16] applied a numerical search technique to provide a unique optimum for closed networks with a single customer class. E. de Souza and Gerla [4] found an optimal load distribution of the open class for mixed open and closed networks using numerical methods. Tantawi and Towsley [12], [13] developed algorithmic techniques for obtaining the optimal load distribution policy in distributed computer systems and star configurations with different Poisson streams of requests arriving at various workstations. Woodside and Tripathi [19] gave an optimal allocation for multiple class queueing networks with statistically identical workstations and file servers. A Vertex-Allocation Theorem for file servers for queueing networks was proved in [14].

The model in this paper can be considered an extension to the model considered in [19]. The earlier model assumed that there were no files shared among the workstations. We prove that when the performance metric is throughput, an optimal allocation is in the vertex subset of the search space. Thus, the search space for the optimal allocation of file servers is reduced to just a vertex set. However as the number of vertices is an exponential function of the number of workstations, it is still difficult to obtain the best workload for each file server by an exhaustive search. Heuristics for choosing an optimal workload for each file server are presented and evaluated by comparing results obtained by searching the set of all feasible solutions. In those cases where we are unable to obtain the optimum allocation, we compare the heuristics to an unachievable upper bound.

Section 2 describes the model, introduces notation and presents the modified Vertex-Allocation Theorem. Section 3 presents three heuristics to determine near optimal allocations for file servers. These heuristics are compared to each other and to the numerical results obtained by searching the entire vertex set in section 4.

2 The Model

We consider a local area network with R workstations and M file servers as shown in Figure 1. Usually, R is much larger than M. Each workstation stores information on the file servers, and executes a two-phase cycle when requesting the information. First, it executes locally for an average duration of Z seconds (including cpu and local disk operations). Second, it accesses a file server requiring, on the average, X seconds of service. The demands at both the workstations and the file servers are statistically identical and independent for all workstations. The communication time is included in the model as part of Z. The contention delays of the communication medium are ignored under the assumption that the network is lightly loaded.

The model is assumed to belong to the separable class of queueing networks [1], that is, the file servers are assumed to have either processor sharing or first-come-first-served (FCFS) discipline. If the FCFS discipline at a file server is assumed, then the service time is assumed to be exponential and each file request has the same mean service time. Last, successive transitions by a job are assumed to be independent.

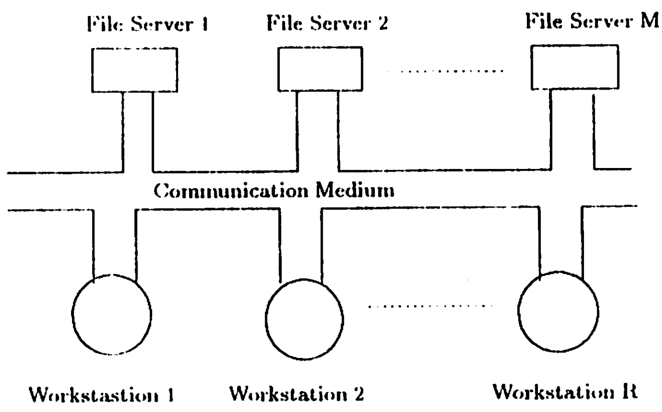

Figure 1: The system

We assume that there is ample storage capacity on the file servers and that the shared files are preallocated to K ($K \leq M$) file servers, which are numbered 1 to K. The remaining file servers are numbered from $K+1$ to M. Workstation r requests the shared files at file server i with probability $\tilde{p}_{ri}$. Let p_{ri} denote the probability that workstation r makes a request to server i where the request can be either to a shared or a private file. Consequently,

$$
\begin{aligned}
0 \leq \ \tilde{p}_{ri} \ &\leq p_{ri} \\
\tilde{p}_{ri} \ &= \ 0, \quad K+1 \leq i \leq M, \qquad 1 \leq r \leq R
\end{aligned}
$$

The system is modeled as a queueing network of M servers and R chains with only one job in each chain (See Figure 2).

The allocation problem is to determine the best values for the probabilities p_{ri}'s that give the optimal performance (discussed later) and satisfy the constraints:

$$\forall r, \quad \tilde{p}_{ri} \leq p_{ri} \leq 1, \quad \text{and} \tag{1}$$

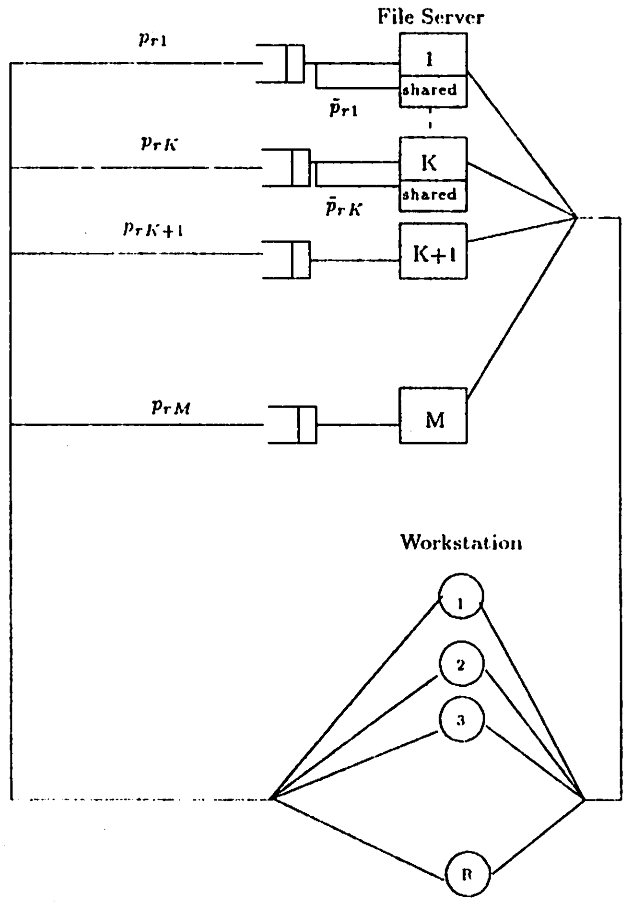

Figure 2: Queueing Network Model

$$\forall r, \quad \sum_{i=1}^{M} p_{ri} = 1. \tag{2}$$

The performance measure that we wish to optimize is the total throughput in file accesses per unit time for the system. Maximizing this performance metric corresponds to maximizing the total file server throughput. Since the service time of each file server is the same as others and the throughput of a file server is directly proportional to its utilization, we are concerned with maximizing the total utilization of all file servers. The total utilization is defined as

$$U(\vec{P}) = \sum_{i=1}^{M} U_i(\vec{P})$$

where $U_i(\vec{P})$ is the utilization of the i^{th} file server for the allocation matrix $\vec{P} = (\vec{p}_1, \vec{p}_2, \vec{p}_3, ..., \vec{p}_R)$ where $\vec{p}_l = (p_{l1}, p_{l2}, ..., p_{lM})$.

We define $\vec{P}$ to be a feasible solution if it satisfies constraints (1) and (2). Let $\mathcal{T}$ denote the set of all feasible allocation vectors, i.e., $\mathcal{T} = \{\vec{P} | \tilde{p}_{ri} \leq p_{ri} \leq 1, \sum_{i=1}^{M} p_{ri} = 1\}$. We define the set of vertex allocations to be $V \subseteq \mathcal{T}$ as:

$$\begin{aligned} V &= \{\vec{P} | \vec{P} \in \mathcal{T},\ p_{ri} \in \{\tilde{p}_{ri}, 1 - \sum_{l \neq i}^{K} \tilde{p}_{rl}\},\ 1 \leq i \leq K\} \\ &\quad \cup \{\vec{P} | \vec{P} \in \mathcal{T}, p_{ri} \in \{\tilde{p}_{ri}, 1 - \sum_{l=1}^{K} \tilde{p}_{rl}\},\ K+1 \leq i \leq M\} \end{aligned}$$

An allocation $\vec{P} \in V$ is called a vertex allocation.

The following theorem, which is called the *Vertex Allocation Theorem*, is based on the fact that the objective function (total utilization/throughput) is a monotonic function (increasing or decreasing) of p_{ri} [15]. Therefore, increasing (or decreasing) the value of p_{ri} does not decrease the objective function. Consequently, the value of p_{ri} can be either increased to its upper bound, $1 - \sum_{l \neq i}^{K} \tilde{p}_{rl}$, or decreased to its lower bound, $\tilde{p}_{ri}$, without decreasing the objective function.

THEOREM 2.1 *For the queueing network defined above, an optimal allocation $\vec{p} \in \mathcal{T}$ can be found in the vertex set V.*

The proof is similar to those used to prove Theorem 1 in [15]. In fact, we can view the shared files as preallocated service centers. This Theorem implies that one of the optimal allocations places all allocatable files in one file server for each chain in the network. The next section describes several heuristics for allocating workstations to file servers.

3 Algorithms for Allocations

If no file is shared ($\tilde{p}_{r1} = \tilde{p}_{r2} = \cdots = \tilde{p}_{rK} = 0$) and all file servers are identical, the best workload for each file server is given by the following theorem [19]:

THEOREM 3.1
If no file is shared ($\tilde{p}_{r1} = \tilde{p}_{r2} = \ldots = \tilde{p}_{rK} = 0$), an optimum allocation for the queueing model assigns s_i workstations to the i^{th} file server, where $s_i = J+1$ for any F file servers and $s_i = J$ for the others, J and F satisfy the following relation:

$$R = JM + F.$$

In this case, the workload for each file server is balanced. If there are shared files, the problem is much more complicated. According to Theorem 2.1, all nonshared files of a workstation should be allocated to one file server. Since there are M file servers and R workstations in the system, there are M^R possible allocations. We can find the optimal allocation by computing the total utilization for each allocation and choosing the allocation with the largest utilization. If we do so, the time required for finding an optimal allocation would be $O(M^R \times T_u)$ where T_u is the time required to solve the multi-class queuing network model. Not only do the number of feasible vertex allocations grow exponentially as a function of the number of workstations, R, but the time required to solve the model for each allocation also grows exponentially as a function of R [11].

As the computational requirements for obtaining an optimal solution to the problem are high, we will focus on heuristics for obtaining good *suboptimal* solutions to the problem. As a first step to the development of these heuristics, we focus on the following new optimization problem:

$$\text{Minimize} \quad \sum_{i=1}^{M} \left| \sum_{r=1}^{R} p_{ri} - \frac{R}{M} \right|$$

subject to

$$\vec{p} \in V.$$

In the context of this new problem we can visualize the file servers as bins into which we are placing fluid (probability). A solution to this new problem attempts to equalize the total amount of fluid in each bin subject to constraint that the allocation be a vertex allocation. One observes that the solution to this problem *no longer requires the solution to a multi-class product form queueing network.*

The new optimization problem is NP complete [9]. The first heuristic for solving this problem is an algorithm that exhaustively searches the set of all feasible vertex allocations V. We shall refer to this as heuristic 1. The complexity of this heuristic is still $O(M^R)$. However, as mentioned earlier, for each allocation we compute the value of $\sum_{i=1}^{M} |\sum_{r=1}^{R} p_{ri} - \frac{R}{M}|$ which takes much less time than computing the total utilization of the system. Hence, the time required for finding an allocation using this heuristic is reduced.

The following heuristic is more efficient than heuristic 1:

Heuristic 2

1. *Initialize* sum_i *for* $i = 1, \cdots, M$,

$$sum_i = \begin{cases} \sum_{r=1}^{R} \tilde{p}_{ri}, & i = 1, \cdots, K, \\ 0, & i = K+1, \cdots, M. \end{cases}$$

2. *For* $r := 1$ *to* R *do*

   (a) *Let* sum_t *be the minimum of* sum_i, i=1 *to* M.

   (b) $sum_t := sum_t + \hat{p}_r$ *(allocate workstation* r *to server* t*)*

where $\hat{p}_r$ *is equal to* $1 - \sum_{i=1}^{M} \hat{p}_{ri}$.

In heuristic 2, all nonshared files of a workstation are allocated to the server with the smallest *sum* at each step. The complexity of this heuristic is $O(RM)$, which is considerably less than the complexity of heuristic 1. This complexity can be further reduced to $O(K \log_2 K + R \log_2 M)$ if an ordered list of file servers according to sum_i's is kept.

For heuristic 3, we first allocate the nonshared files of a workstation to servers $K+1$ to M until the sums of the probabilities exceeds a predefined value, ***threshold***. Then, we allocate the remaining nonshared files to servers 1 to M in a sequential manner. The choice of the ***threshold*** value certainly affects the results of the heuristic 3. In the extreme case if ***threshold*** $= 0$, heuristic 3 is exact the same as the baseline algorithm. On the other hand if we choose a very large ***threshold***, no files will be allocated to shared file servers. To avoid the extreme cases, we let ***threshold*** be the average of sums of $\tilde{p}_{ri}$'s, that is, ***threshold*** $= \frac{\sum_{i=1}^{K}\sum_{r=1}^{R}\tilde{p}_{ri}}{K}$.

Heuristic 3

1. *Initialize sum_i for $i = 1, \cdots, M$,*

$$sum_i := \begin{cases} \sum_{r=1}^{R} \tilde{p}_{ri}, & i = 1, \cdots, K, \\ 0, & i = K+1, \cdots, M. \end{cases}$$

2. *threshold* $:= \frac{\sum_{i=1}^{K} sum_i}{K}$.

3. *count* $:= M - K$; (number of file servers without shared files)

   (a) *WHILE (count > 0) AND ($r \le R$) DO*

   (If there is a sum_t for $K+1 \le t \le M$ which is smaller than the ***threshold***, this step allocates all nonshared files of a workstation to it)

   BEGIN

   - *choose a server, t, from servers $K+1$ to M whose sum_t is smaller than threshold.*
   - $sum_t := sum_t + \tilde{p}_r$;
   - $r := r+1$;

- *If* $sum_t \geq$ *threshold, count := count -1;*

END

(b) *l :=1;*

(c) *WHILE* $r \leq R$ *DO*

(If all sum_i's for $i = K+1, \ldots, M$ are greater than the threshold, allocate the rest workstations to servers $1, \cdots, M$ in turn)

BEGIN

- $sum_l := sum_l + \hat{p}_r$;
- $r := r+1$; (for allocating next workstation)
- $l := l+1$; *If* $l > M$ *then* $l := 1$;

END

The choice of a server t in the first step of (a) can be based either on the NEXT-FIT algorithm, that is, we allocate files of workstations sequentially to a server whose *sum* is less than the *threshold*; or on the WORST-FIT algorithm, that is, we allocate files to the server with the minimum value of *sum*. The worst case complexity of heuristic 3 is $O(R(M-K))$. If an ordered list of sum_i's for $i = K+1, \cdots, M$ is kept when we use the WORST-FIT algorithm, the worst case complexity can be reduced to $O(R\log_2(M-K))$.

It is difficult to theoretically evaluate the quality of these heuristics. In the next section, we will evaluate them empirically.

4 Numerical Validation

In this section we describe the results of a number of tests to evaluate the heuristics presented in the previous section. These tests compare the allocations obtained by the heuristics with each other and with the optimal allocations. The optimal allocations are obtained using a numerical search procedure to search the entire vertex set. This numerical search involves solving the queueing model for all possible parameters using the solution packet QNAP2 [17]. We also perform several experiments where we were unable to obtain the optimal allocation for comparison with the heuristics. In these cases, we obtain an ***unachievable upper bound*** on performance by assuming that *all* files are private and applying the algorithm presented in Theorem 3.1.

We perform two sets of experiments that differ according to the average terminal service time. In one $Z=1$ and in the other $Z=3$. In these experiments, the service times of file servers are all assumed to be exponentially distributed with an average of 1 time unit, $X = 1$. The values of shared probabilities, $\tilde{p}_{ri}$'s, are randomly assigned in each experiment. By applying heuristics 1, 2 and 3 (using both NEXT-FIT and WORST-FIT), we get four allocations. In addition to comparing the three heuristics to the optimal solution, we also compare it to the following baseline algorithm:

Baseline Algorithm:

Randomly assign workstations to file servers so that the total number assigned to each file server differs by at most one.

Table 1 shows the experiments for $Z = 1$. These experiments include models of 3, 4, 5 and 6 workstations with 3 file servers. Only file server 1 has the shared files. From this table, we find that heuristic 1 gives the best results in most experiments. The results from heuristic 2 are the same as or slightly better than those using the two variations of heuristic 3. We also observe that the heuristic 3 using WORST-FIT performs a little bit better than that using NEXT-FIT in some cases. But the differences are very small. All of them are much better than the results obtained by the baseline algorithm.

Tables 2 shows the experiments for $Z = 3$. These tables include three models. The first one is a model of 15 workstations with 4 file servers. The second model is for 4 workstations and 3 file servers. For these two models, there is only one file server with shared files. The third model is for 6 workstations and 3 file servers. In this model, file servers 1 and 2 have shared files. From these tables, heuristic 1 still gives the best results. Heuristic 2 yields slightly better results than both variations of heuristic 3. The performance of all the heuristic allocations is much better than that of the allocation produced by the baseline algorithm.

Tables 3 and 4 show the experiments for a larger number of workstations. Since the size of the vertex allocation set ($O(M^R)$) is quite large, we omit the experiments for numerical search and heuristic 1 and instead provide an upper bound on performance. In table 3 we present results from four experiments for 18 workstations and 3 or 4 file servers. File servers 1 and 2 have shared files with shared probabilities shown in the table. Table 4 also presents results from four experiments. The first one is for 18 workstations and 3 file servers (server 1 has shared files). The second and third are for the 15 workstations and 18 workstations with 3 file servers (two of them have shared files). The last one is for 18 workstations and 4 file servers (two of them have shared files). From these experiments, we observe that the results using the heuristics are all much better than those using the baseline algorithm. Furthermore, heuristic 2 provides performance that is within 4% of the unachievable upper bound in all of these experiments.

Model		Probality	Util	Util	Util	Util	Util
3-3		$\hat{P}_{11}$ =	0.1	0.1	0.5	0.4	0.2
		$\hat{P}_{21}$ =	0.2	0.2	0.3	0.6	0.3
		$\hat{P}_{31}$ =	0.3	0.2	0.8	0.1	0.4
	opt		1.419	1.419	1.286	1.369	1.359
	heu1		1.419	1.419	1.286	1.366	1.348
	heu2		1.419	1.419	1.286	1.344	1.348
	heu3(W)		1.419	1.419	1.286	1.344	1.348
	heu3(N)		1.419	1.419	1.286	1.313	1.337
	base		1.359	1.388	1.195	1.324	1.298
4-3		$\hat{P}_{11}$ =	0.1	0.7	0.1	0.55	0.1
		$\hat{P}_{21}$ =	0.2	0.8	0.15	0.75	0.2
		$\hat{P}_{31}$ =	0.3	0.5	0.2	0.66	0.2
		$\hat{P}_{41}$ =	0.4	0.6	0.25	0.75	0.3
	opt		1.732	1.374	1.746	1.344	1.738
	heu1		1.694	1.374	1.746	1.344	1.738
	heu2		1.694	1.374	1.740	1.344	1.723
	heu3(W)		1.694	1.374	1.731	1.334	1.708
	heu3(N)		1.692	1.373	1.731	1.334	1.738
	base		1.422	1.251	1.546	1.143	1.522
5-3		$\hat{P}_{11}$ =	0.1	0.1	0.1	0.85	0.1
		$\hat{P}_{21}$ =	0.2	0.2	0.25	0.8	0.15
		$\hat{P}_{31}$ =	0.3	0.2	0.15	0.7	0.2
		$\hat{P}_{41}$ =	0.4	0.2	0.2	0.9	0.25
		$\hat{P}_{51}$ =	0.5	0.3	0.3	0.75	0.3
	opt		1.973	2.031	2.034	1.235	2.034
	heu1		1.942	2.031	2.034	1.235	2.034
	heu2		1.973	2.031	2.034	1.235	2.034
	heu3(W)		1.973	2.031	2.034	1.235	2.034
	heu3(N)		1.940	2.031	2.031	1.234	2.033
	base		1.546	1.793	1.816	1.150	1.772
6-3		$\hat{P}_{11}$ =	0.1	0.1	0.1	0.5	0.1
		$\hat{P}_{21}$ =	0.2	0.15	0.15	0.9	0.9
		$\hat{P}_{31}$ =	0.3	0.2	0.2	0.7	0.2
		$\hat{P}_{41}$ =	0.4	0.25	0.2	0.75	0.2
		$\hat{P}_{51}$ =	0.5	0.3	0.05	0.65	0.15
		$\hat{P}_{61}$ =	0.6	0.35	0.25	0.85	0.25
	opt		2.136	2.226	2.262	1.291	2.215
	heu1		2.136	2.226	2.240	1.291	2.215
	heu2		2.135	2.213	2.213	1.291	2.205
	heu3(W)		2.135	2.213	2.224	1.291	2.205
	heu3(N)		2.131	2.207	2.224	1.290	2.191
	base		1.642	1.971	2.176	1.226	2.117

Table 1: Experiments for X=1, Z=1

	15-4		4-3		6-3	
	prob	Util	prob	Util	prob	Util
	$\tilde{p}_{11}=0.1$ $\tilde{p}_{21}=0.1$ ... $\tilde{p}_{15,1}=0.1$		$\tilde{p}_{11}=0.1$ $\tilde{p}_{21}=0.2$ $\tilde{p}_{31}=0.3$ $\tilde{p}_{41}=0.4$		$\tilde{p}_{11}=0.1$ $\tilde{p}_{12}=0.1$ $\tilde{p}_{21}=0.5$ $\tilde{p}_{22}=0.4$ $\tilde{p}_{31}=0.1$ $\tilde{p}_{32}=0.1$ $\tilde{p}_{41}=0.1$ $\tilde{p}_{42}=0.0$ $\tilde{p}_{51}=0.8$ $\tilde{p}_{52}=0.1$ $\tilde{p}_{61}=0.1$ $\tilde{p}_{62}=0.2$	
opt		2.980		0.9605		1.374
heu1		2.980		0.9561		1.366
heu2		2.980		0.9561		1.363
heu3(W)		2.980		0.9561		1.345
heu3(N)		2.980		0.9558		1.345
base		2.901		0.9118		1.330
	$\tilde{p}_{11}=0.2$ $\tilde{p}_{21}=0.2$ ... $\tilde{p}_{15,1}=0.2$		$\tilde{p}_{11}=0.1$ $\tilde{p}_{21}=0.15$ $\tilde{p}_{31}=0.2$ $\tilde{p}_{41}=0.25$		$\tilde{p}_{11}=0.1$ $\tilde{p}_{12}=0.6$ $\tilde{p}_{21}=0.2$ $\tilde{p}_{22}=0.5$ $\tilde{p}_{31}=0.3$ $\tilde{p}_{32}=0.4$ $\tilde{p}_{41}=0.4$ $\tilde{p}_{42}=0.3$ $\tilde{p}_{51}=0.5$ $\tilde{p}_{52}=0.2$ $\tilde{p}_{61}=0.6$ $\tilde{p}_{62}=0.1$	
opt		2.910		0.9635		1.333
heu1		2.910		0.9635		1.332
heu2		2.910		0.9627		1.332
heu3(W)		2.910		0.9616		1.332
heu3(N)		2.910		0.9616		1.332
base		2.626		0.9337		1.285
	$\tilde{p}_{11}=0.3$ $\tilde{p}_{21}=0.3$ ... $\tilde{p}_{15,1}=0.3$		$\tilde{p}_{11}=0.7$ $\tilde{p}_{21}=0.8$ $\tilde{p}_{31}=0.5$ $\tilde{p}_{41}=0.3$		$\tilde{p}_{11}=0.1$ $\tilde{p}_{12}=0.2$ $\tilde{p}_{21}=0.1$ $\tilde{p}_{22}=0.5$ $\tilde{p}_{31}=0.2$ $\tilde{p}_{32}=0.4$ $\tilde{p}_{41}=0.3$ $\tilde{p}_{42}=0.3$ $\tilde{p}_{51}=0.1$ $\tilde{p}_{52}=0.1$ $\tilde{p}_{61}=0.2$ $\tilde{p}_{62}=0.2$	
opt		2.782		0.9037		1.359
heu1		2.782		0.9037		1.359
heu2		2.782		0.9037		1.350
heu3(W)		2.782		0.9037		1.340
heu3(N)		2.782		0.9034		1.340
base		2.268		0.8754		1.314
	$\tilde{p}_{11}=0.5$ $\tilde{p}_{21}=0.5$ ... $\tilde{p}_{15,1}=0.5$		$\tilde{p}_{11}=0.1$ $\tilde{p}_{21}=0.2$ $\tilde{p}_{31}=0.2$ $\tilde{p}_{41}=0.3$		$\tilde{p}_{11}=0.1$ $\tilde{p}_{12}=0.0$ $\tilde{p}_{21}=0.1$ $\tilde{p}_{22}=0.1$ $\tilde{p}_{31}=0.2$ $\tilde{p}_{32}=0.2$ $\tilde{p}_{41}=0.1$ $\tilde{p}_{42}=0.0$ $\tilde{p}_{51}=0.2$ $\tilde{p}_{52}=0.2$ $\tilde{p}_{61}=0.1$ $\tilde{p}_{62}=0.1$	
opt		1.988		0.9625		1.382
heu1		1.988		0.9625		1.381
heu2		1.988		0.9605		1.381
heu3(W)		1.988		0.9584		1.381
heu3(N)		1.988		0.9625		1.381
base		1.665		0.9296		1.365

Table 2: Experiments for X=1, Z=3

	18-3		18-3		18-3		18-4	
shared prob.	$\tilde{p}_{11}=0.2$	$\tilde{p}_{12}=0.1$	$\tilde{p}_{11}=0.5$	$\tilde{p}_{12}=0.4$	$\tilde{p}_{11}=0.2$	$\tilde{p}_{12}=0.1$	$\tilde{p}_{11}=0.1$	$\tilde{p}_{12}=0.0$
	$\tilde{p}_{21}=0.2$	$\tilde{p}_{22}=0.1$	$\tilde{p}_{21}=0.2$	$\tilde{p}_{22}=0.1$	$\tilde{p}_{21}=0.2$	$\tilde{p}_{22}=0.1$	$\tilde{p}_{21}=0.1$	$\tilde{p}_{22}=0.0$
	$\tilde{p}_{31}=0.2$	$\tilde{p}_{32}=0.1$	$\tilde{p}_{31}=0.1$	$\tilde{p}_{32}=0.1$	$\tilde{p}_{31}=0.2$	$\tilde{p}_{32}=0.1$	$\tilde{p}_{31}=0.1$	$\tilde{p}_{32}=0.0$
	$\tilde{p}_{41}=0.2$	$\tilde{p}_{42}=0.1$	$\tilde{p}_{41}=0.1$	$\tilde{p}_{42}=0.1$	$\tilde{p}_{41}=0.1$	$\tilde{p}_{42}=0.1$	$\tilde{p}_{41}=0.0$	$\tilde{p}_{42}=0.1$
	$\tilde{p}_{51}=0.2$	$\tilde{p}_{52}=0.1$	$\tilde{p}_{51}=0.1$	$\tilde{p}_{52}=0.1$	$\tilde{p}_{51}=0.1$	$\tilde{p}_{52}=0.1$	$\tilde{p}_{51}=0.0$	$\tilde{p}_{52}=0.1$
	$\tilde{p}_{61}=0.2$	$\tilde{p}_{62}=0.0$	$\tilde{p}_{61}=0.05$	$\tilde{p}_{62}=0.0$	$\tilde{p}_{61}=0.1$	$\tilde{p}_{62}=0.0$	$\tilde{p}_{61}=0.0$	$\tilde{p}_{62}=0.1$
	$\tilde{p}_{71}=0.2$	$\tilde{p}_{72}=0.0$	$\tilde{p}_{71}=0.2$	$\tilde{p}_{72}=0.0$	$\tilde{p}_{71}=0.2$	$\tilde{p}_{72}=0.0$	$\tilde{p}_{71}=0.1$	$\tilde{p}_{72}=0.1$
	$\tilde{p}_{81}=0.2$	$\tilde{p}_{82}=0.0$	$\tilde{p}_{81}=0.0$	$\tilde{p}_{82}=0.1$	$\tilde{p}_{81}=0.2$	$\tilde{p}_{82}=0.0$	$\tilde{p}_{81}=0.1$	$\tilde{p}_{82}=0.1$
	$\tilde{p}_{91}=0.2$	$\tilde{p}_{92}=0.0$	$\tilde{p}_{91}=0.0$	$\tilde{p}_{92}=0.1$	$\tilde{p}_{91}=0.3$	$\tilde{p}_{92}=0.0$	$\tilde{p}_{91}=0.0$	$\tilde{p}_{92}=0.1$
	$\tilde{p}_{101}=0.2$	$\tilde{p}_{102}=0.0$	$\tilde{p}_{101}=0.2$	$\tilde{p}_{102}=0.0$	$\tilde{p}_{101}=0.3$	$\tilde{p}_{102}=0.0$	$\tilde{p}_{101}=0.0$	$\tilde{p}_{102}=0.2$
	$\tilde{p}_{111}=0.2$	$\tilde{p}_{112}=0.0$	$\tilde{p}_{111}=0.1$	$\tilde{p}_{112}=0.5$	$\tilde{p}_{111}=0.3$	$\tilde{p}_{112}=0.0$	$\tilde{p}_{111}=0.1$	$\tilde{p}_{112}=0.5$
	$\tilde{p}_{121}=0.2$	$\tilde{p}_{122}=0.0$	$\tilde{p}_{121}=0.2$	$\tilde{p}_{122}=0.0$	$\tilde{p}_{121}=0.2$	$\tilde{p}_{122}=0.0$	$\tilde{p}_{121}=0.1$	$\tilde{p}_{122}=0.1$
	$\tilde{p}_{131}=0.2$	$\tilde{p}_{132}=0.0$	$\tilde{p}_{131}=0.1$	$\tilde{p}_{132}=0.0$	$\tilde{p}_{131}=0.2$	$\tilde{p}_{132}=0.1$	$\tilde{p}_{131}=0.1$	$\tilde{p}_{132}=0.5$
	$\tilde{p}_{141}=0.2$	$\tilde{p}_{142}=0.0$	$\tilde{p}_{141}=0.1$	$\tilde{p}_{142}=0.5$	$\tilde{p}_{141}=0.4$	$\tilde{p}_{142}=0.0$	$\tilde{p}_{141}=0.1$	$\tilde{p}_{142}=0.6$
	$\tilde{p}_{151}=0.2$	$\tilde{p}_{152}=0.0$	$\tilde{p}_{151}=0.1$	$\tilde{p}_{152}=0.0$	$\tilde{p}_{151}=0.4$	$\tilde{p}_{152}=0.0$	$\tilde{p}_{151}=0.1$	$\tilde{p}_{152}=0.5$
	$\tilde{p}_{161}=0.2$	$\tilde{p}_{162}=0.0$	$\tilde{p}_{161}=0.2$	$\tilde{p}_{162}=0.0$	$\tilde{p}_{161}=0.4$	$\tilde{p}_{162}=0.0$	$\tilde{p}_{161}=0.1$	$\tilde{p}_{162}=0.5$
	$\tilde{p}_{171}=0.2$	$\tilde{p}_{172}=0.0$	$\tilde{p}_{171}=0.2$	$\tilde{p}_{172}=0.3$	$\tilde{p}_{171}=0.2$	$\tilde{p}_{172}=0.0$	$\tilde{p}_{171}=0.1$	$\tilde{p}_{172}=0.5$
	$\tilde{p}_{181}=0.2$	$\tilde{p}_{182}=0.4$	$\tilde{p}_{181}=0.2$	$\tilde{p}_{182}=0.4$	$\tilde{p}_{181}=0.2$	$\tilde{p}_{182}=0.4$	$\tilde{p}_{181}=0.1$	$\tilde{p}_{182}=0.5$
ub	2.998		2.998		2.998		3.963	
heu2	2.949		2.967		2.933		3.793	
heu3(W)	2.831		2.931		2.770		3.610	
heu3(N)	2.831		2.931		2.770		3.610	
base	2.771		2.860		2.750		2.592	

Table 3: Experiments for $X=1$, $Z=1$

If the probabilities of shared files for all workstations in a file server are equal ($\tilde{p}_{ri} = p_i$ for all r's), heuristic 2 gives the same results as those using heuristic 1 as the result of the following theorem:

THEOREM 4.1
If $\tilde{p}_{ri} = p_i$ for all r's, heuristic 2 and heuristic 1 are identical, i.e., both achieve the minimum value of

$$\sum_{i=1}^{M} \left| \sum_{r=1}^{R} p_{ri} - \frac{R}{M} \right|.$$

PROOF: The proof is based on converting this problem to an integer convex programming problem. Note that heuristic 2 is a variation of Fox's algorithm. We want to know that how many workstations should be allocated to the i^{th} file server [6]. Since all workstations are indistinguishable, let

$$\begin{aligned} p_i &= \tilde{p}_{ri} \quad \text{for } r = 1, \cdots, R \\ p_0 &= 1 - \sum_{i=1}^{K} \tilde{p}_i \quad \text{for } r = 1, \cdots, R \end{aligned}$$

where $\tilde{p}_{ri}$ is the probability of shared files in server i for workstation r.

The problem can be written as

$$\text{Maximize} \sum_{i=1}^{M} f_i(n_i)$$

where n_i is the number of workstations assigned to file server i, and

$$f_i(n_i) = \begin{cases} \frac{R}{M} - Rp_i - n_i p_0, & n_i \le \lfloor \frac{R}{M} - Rp_i \rfloor, \ i \le K \\ n_i p_0 + Rp_i - \frac{R}{M}, & n_i > \lfloor \frac{R}{M} - Rp_i \rfloor, i \le K \\ \frac{R}{M} - n_i p_0, & n_i \le \lfloor \frac{R}{M} \rfloor, \ i > K \\ n_i p_0 - \frac{R}{M}, & n_i > \lfloor \frac{R}{M} \rfloor, \ i > K \end{cases}$$

Note that f_i is convex. Therefore, $-f_i$ is concave. We can apply Fox's algorithm to maximize $-\sum_{i=1}^{M} f_i(n_i)$. Since heuristic 2 is a variation of Fox's algorithm, it yields the optimal solution. □

	18-3	15-3		18-3		18-4	
shared prob.	$\bar{p}_{11}=0.5$	$\bar{p}_{11}=0.5$	$\bar{p}_{12}=0.4$	$\bar{p}_{11}=0.1$	$\bar{p}_{12}=0.0$	$\bar{p}_{11}=0.1$	$\bar{p}_{12}=0.0$
	$\bar{p}_{21}=0.2$	$\bar{p}_{21}=0.2$	$\bar{p}_{22}=0.1$	$\bar{p}_{21}=0.1$	$\bar{p}_{22}=0.0$	$\bar{p}_{21}=0.1$	$\bar{p}_{22}=0.0$
	$\bar{p}_{31}=0.1$	$\bar{p}_{31}=0.1$	$\bar{p}_{32}=0.1$	$\bar{p}_{31}=0.1$	$\bar{p}_{32}=0.0$	$\bar{p}_{31}=0.1$	$\bar{p}_{32}=0.0$
	$\bar{p}_{41}=0.1$	$\bar{p}_{41}=0.1$	$\bar{p}_{42}=0.1$	$\bar{p}_{41}=0.0$	$\bar{p}_{42}=0.0$	$\bar{p}_{41}=0.0$	$\bar{p}_{42}=0.1$
	$\bar{p}_{51}=0.1$	$\bar{p}_{51}=0.1$	$\bar{p}_{52}=0.1$	$\bar{p}_{51}=0.0$	$\bar{p}_{52}=0.1$	$\bar{p}_{51}=0.0$	$\bar{p}_{52}=0.1$
	$\bar{p}_{61}=0.5$	$\bar{p}_{61}=0.5$	$\bar{p}_{62}=0.0$	$\bar{p}_{61}=0.0$	$\bar{p}_{62}=0.1$	$\bar{p}_{61}=0.0$	$\bar{p}_{62}=0.1$
	$\bar{p}_{71}=0.2$	$\bar{p}_{71}=0.2$	$\bar{p}_{72}=0.0$	$\bar{p}_{71}=0.1$	$\bar{p}_{72}=0.11$	$\bar{p}_{71}=0.1$	$\bar{p}_{72}=0.1$
	$\bar{p}_{81}=0.0$	$\bar{p}_{81}=0.0$	$\bar{p}_{82}=0.1$	$\bar{p}_{81}=0.1$	$\bar{p}_{82}=0.12$	$\bar{p}_{81}=0.1$	$\bar{p}_{82}=0.1$
	$\bar{p}_{91}=0.0$	$\bar{p}_{91}=0.0$	$\bar{p}_{92}=0.1$	$\bar{p}_{91}=0.0$	$\bar{p}_{92}=0.13$	$\bar{p}_{91}=0.0$	$\bar{p}_{92}=0.1$
	$\bar{p}_{101}=0.0$	$\bar{p}_{101}=0.2$	$\bar{p}_{102}=0.0$	$\bar{p}_{101}=0.0$	$\bar{p}_{102}=0.2$	$\bar{p}_{101}=0.0$	$\bar{p}_{102}=0.2$
	$\bar{p}_{111}=0.2$	$\bar{p}_{111}=0.1$	$\bar{p}_{112}=0.5$	$\bar{p}_{111}=0.1$	$\bar{p}_{112}=0.5$	$\bar{p}_{111}=0.1$	$\bar{p}_{112}=0.5$
	$\bar{p}_{121}=0.2$	$\bar{p}_{121}=0.2$	$\bar{p}_{122}=0.0$	$\bar{p}_{121}=0.1$	$\bar{p}_{122}=0.1$	$\bar{p}_{121}=0.1$	$\bar{p}_{122}=0.1$
	$\bar{p}_{131}=0.2$	$\bar{p}_{131}=0.2$	$\bar{p}_{132}=0.0$	$\bar{p}_{131}=0.1$	$\bar{p}_{132}=0.0$	$\bar{p}_{131}=0.5$	$\bar{p}_{132}=0.1$
	$\bar{p}_{141}=0.2$	$\bar{p}_{141}=0.2$	$\bar{p}_{142}=0.3$	$\bar{p}_{141}=0.1$	$\bar{p}_{142}=0.0$	$\bar{p}_{141}=0.6$	$\bar{p}_{142}=0.1$
	$\bar{p}_{151}=0.2$	$\bar{p}_{151}=0.2$	$\bar{p}_{152}=0.4$	$\bar{p}_{151}=0.1$	$\bar{p}_{152}=0.5$	$\bar{p}_{151}=0.1$	$\bar{p}_{152}=0.5$
	$\bar{p}_{161}=0.2$			$\bar{p}_{161}=0.1$	$\bar{p}_{162}=0.5$	$\bar{p}_{161}=0.5$	$\bar{p}_{162}=0.1$
	$\bar{p}_{171}=0.2$			$\bar{p}_{171}=0.1$	$\bar{p}_{172}=0.5$	$\bar{p}_{171}=0.1$	$\bar{p}_{172}=0.5$
	$\bar{p}_{181}=0.2$			$\bar{p}_{181}=0.1$	$\bar{p}_{182}=0.5$	$\bar{p}_{181}=0.5$	$\bar{p}_{182}=0.1$
ub	2.998	2.991		2.998		3.963	
heu2	2.979	2.926		2.983		3.850	
heu3(W)	2.979	2.886		2.965		3.824	
heu3(N)	2.977	2.886		2.965		3.824	
base	2.879	2.674		2.602		2.665	

Table 4: Experiments for $X=1$, $Z=1$

We ran a series of models of 4, 9 and 15 workstations with 3, 4 and 5 file servers for both cases of $Z = 1$ and $Z = 3$ where we let $\tilde{p}_{r1}$ take values from 0.1 to 0.9 in increments of 0.1 and $\tilde{p}_{ri} = 0$ for $i = 2, \cdots, M$. The heuristics and OPT deviated in only four cases. The differences are all less than 1%.

5 Conclusion

The modified *Vertex Allocation Theorem* in the section 2 shows that if all shared files are preallocated, all the nonshared files of a workstation should be allocated to one file server. Using this theorem, the search space for an optimal allocation for file servers is reduced to only their vertex space. In section 3, we propose three heuristics to allocate the nonshared files of workstations to file servers. From the experiments shown in the last section, we find that heuristic 1 gives the best results in most cases; heuristic 2 yields allocations that are identical to or slightly better than those using heuristic 3. The differences between the allocations produced by heuristics 1, 2 and 3 and the optimal allocations are very small so as to be insignificant. The allocations using these three heuristics are all much better than those using the baseline algorithm. If the probabilities of shared files in a file server are equal, it can be proved that heuristic 2 is as good as heuristic 1. In this case, the results by heuristic 2 and heuristic 3 coincide with those by numerical search in almost all experiments.

References

[1] F. Baskett, K.M. Chandy, R.R. Muntz, and F. Palacios, Open, closed and mixed networks of queues with different classes of customers. *J. ACM 22,* 2(April 1975), 248-260.

[2] D.P. Bertsekas, E.M. Gafni, and R.G. Gallager, Second Derivative Algorithms for Minimum Delay Distributed Routing in Networks. *IEEE Transactions on Communications,* Vol Com-32, No. 8, 1984, 911-920.

[3] J.P. Buzen, Queueing network models of multiprogramming, Ph.D. Diss., Division of Engineering and Applied Physics, Harvard University, Cambridge, Mass. 1971.

[4] E. de Souza e Silva and M. Gerla, Load balancing in distributed systems with multiple classes and site constraints. *Performance 84,* E. Gelenbe (ed.), North-Holland, 1984, 17-33.

[5] L.W. Dowdy, D.V. Foster, D.B. McCormick, J.F. Nixon, A. Perez-Davila, and L.E. Stephens, Dynamic File Assignment for Local Area Networks: Models and Analysis. Department of Computer Science, Vanderbilt University, Nashville, TN., Tech. Rep. CS-84-06, 1984.

[6] B. Fox, Discrete Optimization via Marginal Analysis. *Management Science 13,* (1966) 212-216.

[7] L. Fratta, M. Gerla, and L. Kleinrock, The flow deviation method - an approach to store-and-forward communication network design. *Networks 3,* 1973.

[8] R.G. Gallager, A Minimum Delay Routing Algorithm Using Distributed Computation. *IEEE Transactions on Communication,* Vol. Com-25, No. 1, 1977, 73-85.

[9] Michael R. Garey and David S. Johnson, *Computers and Intractability*, W. H. Freeman and Company, San Francisco, 1979.

[10] H. Kobayashi and M. Gerla, Optimal routing in closed queueing networks. *ACM Tran. on Computer Systems 1*, 4(1983), 294-310.

[11] S.S. Lavenberg (ed.), *Computer Performance Modeling Handbook*. Academic Press, 1983.

[12] A.N. Tantawi and D. Towsley, A general model for optimal static load balancing in star network configurations. *Performance 84*, E. Gelenbe (ed.), North-Holland, 1984, 277-281.

[13] A.N. Tantawi and D. Towsley, Optimal static load balancing in distributed computer networks. *Journal of ACM*, 32, 2 (April 1985), 445-465.

[14] S.K. Tripathi and L.W. Dowdy, Workload Representation and its Impact on the Performance Prediction Using Queueing Network Models. in *Workload Characterization of Computer Systems and Computer Networks*, G. Serazzi (editor), Elsevier Science Publishers B.V. (North-Holland), 1986, 159-178.

[15] S.K. Tripathi and C.M. Woodside, A Vertex-Allocation Theorem for Resources in Queuing Networks. *JACM, 35, 1 (January 88), 221-230.*

[16] K.S. Trivedi, R.A. Wagner, and T.M. Sigmon, Optimal selection of cpu speed, device capacities, and file assignments. *J. ACM 27*, 3 (1980), 457-473.

[17] M. Veran and D. Potier, "QNAP2: A Portable Environment for Queueing Systems Modeling", *International Conference on Modeling Techniques and Tools for Performance Analysis*, Paris, May 1984.

[18] P. Whittle, Optimal routing in Jackson Networks, *Asia Pacific J. Op. Res.*, No. 1, 1984.

[19] C.M. Woodside and S.K. Tripathi, Optimal allocation of file servers in a local network environment. *IEEE Transactions on Software Engineering,* Vol SE-12, No. 8, 1986, 844-848.