A Decomposition Solution to a Queueing Network Model of a Distributed File System with Dynamic Locking *

Anna Hać **

Computer Science Division
Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley

ABSTRACT

This paper presents a new approach to modeling file systems using queueing networks. The delays due to the locking of the files are modeled using service centers whose service times and probabilities of access are estimated from the values of measurable quantities. The model of a lock is based on the analysis of the execution of transactions in the system. The lock for every file is modeled as a sequence of service centers. The decomposition method can be used to solve the model, which allows multiple classes of transactions and shared files to be represented. An example involving measurement data collected in a small business installation is given to compare performance measures provided by simulation and analytic models.

1. Introduction

Modeling distributed file systems requires sophisticated methods of analysis due to the complex nature and behavior of such systems.

The methodology for selecting a file system organization presented in [4] includes an algorithm for distributing shared files over hosts, and the application of queueing network models for distributed systems. In [4], the methodology is also applied to a simple example involving workload data measured in a small business installation.

Another paper [6] presents a distributed system model, which is solved using mean value analysis. In that paper, the performance predictions of the model were validated using Locus, a distributed operating system [8, 10].

This paper focuses on a methodology for modeling file systems with dynamic locking. The communication network is assumed to be a high-speed local area network [3]. A queueing network model [1, 7] is employed to represent lock behavior [5, 2].

1.1. Motivation and Approach

Let us consider a simple system, whose model is shown in Fig. 1. In this system the terminals initiate interactions which are then sent to the CPU and the file disk, and then return to the terminals. The file disk can have many files on it, and these files can be locked by transactions which access them.

* The research reported here has been supported in part by the State of California and the NCR Corporation under MICRO Grant No. 1-532436-19900.

** On leave from Institute of Computer Science, Technical University of Warsaw, Poland.

When a transaction locks a file and is then sent to the other service centers without releasing the lock, then the time spent in the system by this transaction is given by the response time of the system seen below the line AB (i.e., the time spent by a transaction moving from B to A) in the model in Fig. 1. This response time can be calculated, and then incorporated as a delay server to represent the delay experienced by transactions which require access to a locked file. The model of a system with a delay server is shown in Fig. 2. However, because of the additional server representing the delay, the response time of the system (seen below line AB) is changed, and the service time of the delay server has to be recalculated iteratively. Moreover, if the file is still kept locked, after accessing the disk for a second time, then the delay server must be replaced and its service time must be recalculated iteratively, again. This solution applies only to the case of single class transactions (this also means that the files are shared only by transactions of the same type and not by transactions of different types).

The approach taken in this paper is to model the behavior of a transaction in the system (seen below line AB) while the transactions wait to acquire their locks. The method is non-iterative and allows multiple classes of transactions and files shared by different types of transactions to be represented.

The delays in the system due to locking are modeled using service centers where service times and the probabilities of access are estimated from the values of measurable quantities. The model of a lock is based on the analysis of the execution of transactions in the system. The lock for every file is modeled as a sequence of service centers. The service times of these service centers depend on the loads on all service centers. The model is solved using the decomposition method.

The model can be applied to the case of multiple classes and to that of a multiprocessor or multicomputer system.

As an example, simulation and analytic models will be constructed for a file disk to represent the delays arising when the files are locked by the transactions which are then sent to the other service centers without releasing the lock.

The performance degradation due to locking will be discussed. The values of the performance measures obtained from simulation and analytic models will be compared using measurement data collected in a small business installation.

In Section 2, a model of a distributed file system is described, and the model assumptions are introduced. A model of delays due to locking the files is presented in Section 3. In Section 4, the branching probabilities are calculated for the multiple classes of transactions leaving a disk server and a remote host. An approximate model of a distributed file system is described in Section 5. The model is solved in Section 6 by hierarchical decomposition. The results obtained from simulation and analytic models are presented in Section 7. Finally, Section 8 concludes the paper with a brief description of the problems which remain to be investigated.

2. A Queueing Network Model of a Distributed File System

The system considered in our approach is a distributed file system containing several identical hosts. Each host includes a number of terminals which initiate the processing of transactions. The behavior of a transaction is as follows: each transaction has to be processed in the CPU before it accesses any other service center; it may also be processed by a file disk and by the network server; and it may be delayed because of locks. In its local host a transaction also produces several display outputs before returning to a user terminal.

The model of a single host and that of the multiple host system are shown in Fig. 3 and 4, respectively. The model of each host represents a number of terminals and display outputs, both modeled as infinite servers (is), a CPU modeled as a processor-sharing (ps) server, and a number of file disks, each disk drive being modeled as a first come first served (fcfs) server. Each file disk consists of a directory and a number of files. The directory is located on the same disk as the files and is modeled in the same way as a file. The CPU overhead due to disk accesses is incorporated into the CPU service time.

The workload of the system consists of a number of transactions. It is assumed that each type of transaction defines its own class (i.e., the number of classes is equal to the number of transaction types). Each class of transactions is characterized by the names of files locked and unlocked for each disk access; and, for every file, the number of physical disk accesses done while keeping a file locked.

The following assumptions are made:

(1) a transaction does not change its class within a host;

(2) each file is either locked or non-locked in its entirety;

(3) a transaction unlocks all the files it has locked when leaving a host;

(4) a directory is unlocked immediately after the disk on which it is stored has been accessed (i.e., there are no delays due to locking the directory);

(5) all locks are exclusive (i.e., there are only write locks, not read locks);

(6) all transactions from remote hosts create a separate class in the local host; these transactions are statistically identical in a host and, after leaving the host, the returning transactions are distributed to their original sites probabilistically rather than deterministically, as will be shown in Section 4 (this assumption is made to simplify the model, and is based on the assumption that transactions accessing remote files are statistically identical in the host; otherwise, each transaction type from a remote host should be represented by a separate class in each host).

3. Approximation of Locking Delays

Our model of a locking delay is shown in Fig. 5. The service centers and their queues represent the delay experienced by transactions of one type accessing a single file. This file is locked by a transaction which has then been sent to the other service centers in the model of a host (Fig. 3) without releasing the lock. The transactions waiting in the queue to the disk server have already acquired their locks. The CPU DELAY server and its queue represent the delay experienced by transactions which require access to the file at a time when a transaction which locked it is being executed in the CPU. The mean service time of the CPU DELAY server is estimated as the mean remaining time spent in the CPU by a transaction which locked the file. The DISPLAY DELAY server and its queue represent the delay experienced by transactions which require access to the file at a time when a transaction which locked it is outputting characters on a display screen. The mean service time of the DISPLAY DELAY server is estimated as the mean remaining time spent in the display server by a transaction which locked the file. The TERMINAL DELAY server and its queue represent the delay experienced by transactions which require access to the file at a time when a transaction which locked it is at a terminal. The mean service time of the TERMINAL DELAY server is estimated as the mean remaining time spent in a terminal by a transaction which locked the file. The direct path models the case in which the file is found to be unlocked. The probabilities of accessing the service centers representing the delays are the same as those in the model of a host. The mean service times of these servers will be calculated later in this section. The servers representing delays are modeled as first come first served (fcfs) servers because every transaction of the same class experiences the same mean delay due to finding a file locked by a transaction which is then sent to the other service centers without releasing the lock. This mean delay (i.e., the total amount of time spent in the servers representing delays) is the time a transaction waits to acquire its locks.

In the case of multiple classes (many types of transactions) and many files on a disk accessed by transactions, the delay subnetwork shown in Fig. 5 is repeated.

If a transaction type locks several files on the same disk, then the delays for every file are repeated sequentially.

If several transaction types share a file, then, in addition, the delay for this file is repeated sequentially for every transaction type accessing it, and is accessed by all transaction types which share it.

For instance, the case when transactions of class 1, 2 and 3 access files 1, 2 and 3, respectively, is shown in Fig. 6. Parallel paths model the disk accesses by transactions of class 1, 2 and 3 (i.e., the probabilities of accessing delays 11, 22 and 33 are equal to the probabilities of accessing the disk by transactions of class 1, 2 and 3, respectively, in the model of a host). The mean service times of the servers in the model of the delays are given by relationships (1), (2) and (3) (relationships (1), (2) and (3) will be given later in this section), where i is equal to 1, 2 and 3 for delays 11, 22 and 33, respectively.

The case where the second file is shared by transactions of classes 1 and 2 is shown in Fig. 7. Parallel paths model disk accesses by transactions of class 1, 2 and 3 (i.e., the probabilities of accessing delays 11, 22 and 33 are equal to the probabilities of accessing the disk by transactions of class 1, 2 and 3, respectively, in the model of a host). The mean service times of the servers in the model of the delays are given by relationships (1), (2) and (3), where i is equal to

(a) 1 for delays 11 and 12;

(b) 2 for delay 22;

(c) 3 for delay 33.

The delays 12 and 22 are repeated for transactions of class 1 and 2, since these transactions share the second file. The reason is that transactions of class 1 and 2 compete for access to the second file not only with transactions of the same class but also with transactions of the other class.

When there are many file disks, then the delays for transactions accessing the files are modeled in the same way for every file disk.

The probability p_i of experiencing a delay when accessing a file is estimated as the ratio of the number of disk accesses made while keeping the file locked to the total number of disk accesses of a transaction of class i.

The mean service times of the service centers in the model of a delay are given by:

$$s_i^{CPU} = min\Big(max\Big((n-n_i)\frac{p_i}{1-p_i}\Big(\sum_{k=1}^{nc,i} t_k^{cpu} - pdisp_i\frac{t_i^{disp}}{(n_i-1)(n-n_i)} - pterm_i\frac{t_i^{term}}{(n_i-1)(n-n_i)}\Big), \quad (1)$$

$$\sum_{k=1}^{nc,i} t_k^{cpu}\frac{n_i-1}{n(n-n_i)}\Big), \sum_{k=1}^{nc,i} t_k^{cpu}\Big),$$

$$s_i^{DISPLAY} = min\Big(max\Big(\frac{p_i}{1-p_i}\Big(pdisp_i t_i^{disp} - t_i^{cpu} - pterm_i\frac{t_i^{term}}{n_i-1}\Big), \quad (2)$$

$$pdisp_i t_i^{disp}\frac{n_i-1}{n(n-n_i)}\Big), t_i^{disp}\Big),$$

$$s_i^{TERMINAL} = min\Big(max\Big(\frac{p_i}{1-p_i}\Big(pterm_i t_i^{term} - t_i^{cpu} - pdisp_i\frac{t_i^{disp}}{n_i-1}\Big), \quad (3)$$

$$pterm_i t_i^{term}\frac{n_i-1}{n(n-n_i)}\Big), t_i^{term}\Big),$$

where t_i^{cpu}, t_i^{disp} and t_i^{term} are the mean service time of the CPU, the mean time of a display output, and the mean user "think" time, respectively, for a transaction of class i; n_i is the number of transactions of class i; nc is the number of classes of transactions which access files placed on the different disks than the files accessed by transaction of class i (if there are many file disks), and do not share files on these disks (i.e., all the classes of transactions which share files on any disk are considered as a single class in the calculation of nc; this is why the index k goes from 1 to nc and then becomes equal to i since i is not between 1 and nc); and $pdisp_i$ and $pterm_i$ are the probabilities that a transaction of class i is sent to a display or to a user terminal in the model of a host.

The service times of the service centers in the model of a delay are limited by the service time of the CPU, the time of a display output and the user "think" time at a terminal (relationships (1), (2) and (3), respectively).

The displays and the terminals are modeled as infinite servers in the model of a host, and there is no queueing at these servers; this is why a transaction sent to these servers spends there no more time then their service time.

The term $\sum_{k=1}^{nc,i} t_k^{cpu}$ in (1) is the time that a transaction of class i spends in the CPU, in the model of a host, if $nc+1$ transactions queue at the CPU.

Let us analyze the limitation in equation (1).

If a transaction of class i locks a file and is then sent to the other service centers without releasing the lock, then all transactions of class i (after a certain amount of time) queue to obtain the lock and can no longer compete for access to the CPU. This also applies to all transactions which share any file locked by a transaction of class i.

So, a transaction of class i (which locked a file) is executed in the system concurrently only with transactions which lock other files on the same disk or files on other disks.

However, since the servicing of requests in the CPU and in a disk proceed in parallel, and all transactions have to queue at the disk server after acquiring their locks, then the delay due to the locking of a file depends only on the behavior of transactions of class i and of the nc types of transactions which access different disks On the other hand, the maximum number of transactions proceeded in the system is equal to $nc+1$, since only one transaction per every type is executed and all other transactions of class i and of the nc classes of transactions wait to acquire their locks and do not compete for access to the other resources.

This is why the longest time that a transaction of class i spends in the CPU, in the model of a host, is equal to $\sum_{k=1}^{nc,i} t_k^{cpu}$ when $nc+1$ transactions (i.e., one transaction per type) queue at the CPU.

4. Approximation of Branching Probabilities

Since the servers in the model of the delay and also in that of the disk are assumed to be first come first served (fcfs) servers, the various classes of transactions cannot be distinguished in these service centers. The probabilities of accessing the CPU from the file disk are calculated below.

Let us define

$$c_i = p_i \left(t_i^{cpu} + \frac{1-pdisk_i}{pdisk_i}\left(t_i^{cpu} + pdisp_i \frac{t_i^{disp}}{n_i(n-n_i)} + pterm_i \frac{t_i^{term}}{n_i(n-n_i)}\right)\right), \tag{4}$$

where $\frac{1-pdisk_i}{pdisk_i}$ is the number of times that a transaction of class i is sent to the CPU (and possibly to a display and a terminal) before returning to the disk.

The probability of accessing the CPU after having accessed the disk server for a transaction of class i is given by

$$pr_i^{cpu} = \frac{\frac{n_i}{c_i}}{\sum_{i=1}^{ncd} \frac{n_i}{c_i}}, \tag{5}$$

where ncd is the number of classes of transactions accessing this disk server.

A similar approach is used to direct transactions to their original sites when they leave a remote host. Let pn_{jk} be the probability of accessing host j by a transaction of class i on host k, n_{ik} be the number of active transactions of class i on host k, and nch be the number of classes of local transactions on host k. The number of active transactions accessing host j from host k (because of the assumption that all transactions from the remote hosts create a separate class in a

given host, k is considered as a class) is given by

$$n_{jk} = \sum_{i=1}^{nch} pn_{jki}\, n_{ki}. \tag{6}$$

Let ph_j be the probability that host j be exited by transactions being executed temporarily on j, and NH be the number of hosts. The probability of returning to host k is given by

$$p_k^{host} = \frac{n_{jk} ph_j}{\sum_{m=1}^{NH} n_{jm} ph_j}. \tag{7}$$

5. Approximation of a Model of a Distributed File System

Another approximate model was constructed for the same distributed system. The representation of a single host in this model is shown in Fig. 8. The model of the multiple host system is the same as that shown in Fig. 4. The host is represented by a CPU, a number of disk servers, the servers describing the delays due to locks modeled in the same way as described in Section 3 and shown in Fig. 5, and two additional servers (i.e., displays and terminals). These two additional servers represent the delays in processing transactions on this host. These delays are introduced by users working within this host.

The probabilities of accessing the service centers in the model in Fig. 8 are the same as in the model in Fig. 3.

The reason for this representation of a host is to simplify (or even make possible) mathematical analysis.

The mean service times of the servers representing delays in processing transactions on a host are given by:

$$s^{DISPLAYS} = min\Big(max\Big(\frac{1}{nch}\sum_{i=1}^{nch}\Big(pdisp_i t_i^{disp} - t_i^{cpu} - pterm_i \frac{t_i^{term}}{n_i}\Big), \tag{8}$$

$$\frac{1}{nch}\sum_{i=1}^{nch} pdisp_i t_i^{disp}\frac{n_i}{n(n-n_i)}\Big), \frac{1}{nch}\sum_{i=1}^{nch} t_i^{disp}\Big),$$

$$s^{TERMINALS} = min\Big(max\Big(\frac{1}{nch}\sum_{i=1}^{nch}\Big(pterm_i t_i^{term} - t_i^{cpu} - pdisp_i \frac{t_i^{disp}}{n_i}\Big), \tag{9}$$

$$\frac{1}{nch}\sum_{i=1}^{nch} pterm_i t_i^{term}\frac{n_i}{n(n-n_i)}\Big), \frac{1}{nch}\sum_{i=1}^{nch} t_i^{term}\Big),$$

where nch is the number of classes of local transactions in the model in Fig. 8.

These servers replace displays and terminals in the model in Fig. 3. The reason to model them as first come first served servers is to represent delays in processing transactions on a host in the same way as locking delays and to solve the model using the algorithm given in the next section.

6. Decomposition Solution

The model shown in Figs. 3 and 4 can be solved by hierarchical decomposition. We shall consider three successive submodels of it.

The first submodel represents a disk and the delays for accessing the files placed on it. The service times of the service centers in the subnetworks representing delays are calculated using relationships (1), (2) and (3).

The second submodel represents the host as a central server model; each file disk is represented by a service center flow-equivalent to the first submodel.

The third submodel, which encompasses the whole model, consists of a network server and a number of hosts; each host is represented by a service center flow-equivalent to the second submodel.

The evaluation of the performance measures for the submodels and for the outer model will now be described for each transaction type.

For the second submodel the performance measures are calculated using a computational algorithm based on Polya's theory of enumeration, an application of group theory to combinatorial problems [7]. The algorithm evaluates the normalization constant $\psi(N,M)$ and is restricted to networks of exponential servers with fixed service rates. Our submodel is evaluated in a steady state for fixed number of users.

Let M be the number of service centers and N be the number of users in the submodel. If service center i has a constant service rate τ_i, then the utilization of this service center is given by

$$\rho_i = \tau_i \frac{\psi(N-1,M)}{\psi(N,M)} . \tag{10}$$

For each host in the model, the average service times s_i for all service centers can be calculated using the access probabilities as the weighting factors. For the submodel representing a file disk, the mean service time is given by

$$s_{submodel}^{file\_disk} = s_{disk} + s_{delay} , \tag{11}$$

where s_{disk} and s_{delay} represent the mean service time of the disk server, and the mean service time of the delay server, respectively.

The utilization of each service center is given by (10), where M is the number of service centers within a host, and N is equal to the number of parallel transactions executed within the host (i.e., the degree of multiprogramming).

In the third submodel, each of the servers is equivalent to the second submodel. The service rates of these servers are not constant. They depend on the number of multiprogrammed transactions, and are given by

$$\mu(n) = \frac{\psi(N-1,M)}{\psi(N,M)} , \tag{12}$$

where n is the number of transactions within the local host (i.e., of the transactions that are multiprogrammed plus those waiting in the queue to the CPU in the second submodel).

Approximating the system service rate function of the server of the third submodel by averaging over n, as

$$\mu(n) = \overline{\mu(n)} , \tag{13}$$

and knowing the transaction generation rate $\nu = \frac{1}{T_{TH}}$ at a user terminal, the mean response time is given by

$$\overline{T} = \frac{n - \frac{\overline{\mu(n)}}{\nu}}{\overline{\mu(n)}} = \frac{n}{\overline{\mu(n)}} - \frac{1}{\nu} . \tag{14}$$

The system throughput is given by

$$\lambda = \rho \overline{\mu(n)} , \tag{15}$$

where ρ is the system utilization factor.

The decomposition solution of our approximate model of a distributed system gives results, which will be compared with the results of the non-approximate analytic model in the next section.

7. Some Results

Two simple examples of performance prediction (approached with the RESQ2 queueing network analysis package [9]) in centralized and distributed systems using measurements collected in a small business installation will now be discussed to show an application of the models and to evaluate the performance degradation due to locking.

The workload of the system is defined by five types of transactions. The following data are measured for each transaction type:
I - the number of interactions;
T_{TH} - the mean "think" time of an interaction;
T_S - the mean time of a display output;
T_{CPU} - the mean CPU time consumed by a transaction of that type;
D - the total number of disk I/O operations in a transaction of that type;
S - the total number of display outputs in a transaction of that type;
F - the number of files accessed by a transaction of that type;
ND - the number of disk I/O operations done while keeping a file locked, for every file;
P - the percentage of the number of transactions of that type in the workload.

The following parameter values can be derived from the measurement data:

Average CPU time per interaction $(\frac{T_{CPU}}{I})$

Average number of disk I/O operations per interaction $(\frac{D}{I})$

Average number of display outputs per interaction $(\frac{S}{I})$

Average CPU time between successive disk I/O operations and/or display outputs $(\frac{T_{CPU}}{D+S})$

Probability of accessing the file disk $(\frac{D}{1+D+S})$

Probability of experiencing a locking delay for each file $(\frac{ND}{D})$

Probability of a display output $(\frac{S}{1+D+S})$

The estimated values of the parameters of each transaction type (i.e., each class in the model) are given in Tables I and II for single host system and for a system with two hosts, respectively. In Table II only the values of parameters different from those in Table I are shown. Also, the probability of accessing a remote host (P_H) is derived assuming that a number of the disk I/O operations for each transaction type is performed in the remote host. The probability of accessing a file disk (P_F) by a remote transaction is calculated on the basis of the number of local accesses made by this transaction. Note that the remote transactions do not do remote display outputs and do not return to a user terminal in the remote host.

Table I. Estimated values of the model parameters for a single host system.

Transaction Type	$T_{CPU}/(D+S)$ [msec]	T_{TH} [msec]	T_S [msec]	$D/(1+D+S)$	$S/(1+D+S)$
1	42	1000	15	0.4	0.49
2	35	10000	25	0.8	0.19
3	224	1000	20	0.75	0.2
4	32	500	15	0.966	0.027
5	15	2000	15	0.3	0.61

Table II. Estimated values of the model parameters for a system of two hosts.

Transaction Type	D/(1+ D+ S+ H)	S/(1+ D+ S+ H)	P_H	P_F
1	0.39	0.47	0.04	0.79
2	0.74	0.17	0.07	0.987
3	0.7	0.19	0.07	0.943
4	0.88	0.025	0.088	0.993
5	0.29	0.59	0.03	0.778

The average service times for the following two service centers are the same for all transaction types:

Disk access $T_{DISK} = 30$ *msec*
Network server $T_{NET} = 0.2$ *msec*

In the first example, the performance measures are computed for a centralized system containing one file disk with three files on it. The system was modeled using simulation and numerical methods. The confidence level of the simulation was 90 percent. The confidence interval width was (-10, + 10) percent. Performance measures were calculated for the simulation and analytic models for the no-sharing case (as shown in Fig. 6), and when one file is shared (as shown in Fig. 7). The results of the comparison of the models are presented in Tables III , IV, V and VI.

Table III. Performance measures for the simulation and analytic models when no file is shared by transaction types.

Model*	Transaction Types	Number of Transactions of Each Class	Probability of Delay	Performance Measures			
				Utilization		Throughput [1/sec]	Response Time [sec]
				CPU	Disk		
SIM	1; 2; 5	2; 2; 2	0.1	0.78	0.38	1.75	1.52
AN	1; 2; 5	2; 2; 2	0.1	0.75	0.39	1.78	1.38
SIM	1; 2; 5	2; 2; 2	0.25	0.73	0.36	1.60	1.85
AN	1; 2; 5	2; 2; 2	0.25	0.74	0.38	1.75	1.43
SIM	1; 2; 5	2; 2; 2	0.5	0.65	0.32	1.44	2.28
AN	1; 2; 5	2; 2; 2	0.5	0.67	0.35	1.58	1.81

* SIM - simulation, AN - analytic

Table IV. Performance measures for each type of transactions for the simulation and analytic models when no file is shared by transaction types.

Model*	Probability of Delay	CPU Utilization for Transactions of Type			Throughput for Transactions of Type		
		1	2	5	1	2	5
SIM	0.1	0.35	0.29	0.13	0.95	0.09	0.70
AN	0.1	0.28	0.31	0.16	0.73	0.09	0.95
SIM	0.25	0.32	0.29	0.12	0.87	0.08	0.65
AN	0.25	0.27	0.31	0.16	0.72	0.09	0.93
SIM	0.5	0.29	0.25	0.11	0.78	0.07	0.57
AN	0.5	0.25	0.28	0.14	0.65	0.08	0.84

* SIM - simulation, AN - analytic

Table V. Performance measures for the simulation and analytic models when one file is shared by transaction types.

Model*	Transaction Types	Number of Transactions of Each Class	Probability of Delay	Performance Measures			
				Utilization		Throughput [1/sec]	Response Time [sec]
				CPU	Disk		
SIM	1; 2; 5	2; 2; 2	0.1	0.72	0.35	1.66	1.81
AN	1; 2; 5	2; 2; 2	0.1	0.75	0.39	1.77	1.40
SIM	1; 2; 5	2; 2; 2	0.25	0.60	0.30	1.47	2.25
AN	1; 2; 5	2; 2; 2	0.25	0.71	0.37	1.68	1.60
SIM	1; 2; 5	2; 2; 2	0.5	0.45	0.22	1.15	3.29
AN	1; 2; 5	2; 2; 2	0.5	0.51	0.27	1.22	2.93

* SIM - simulation, AN - analytic

Table VI. Performance measures for each type of transactions for the simulation and analytic models when one file is shared by transaction types.

Model*	Probability of Delay	CPU Utilization for Transactions of Type			Throughput for Transactions of Type		
		1	2	5	1	2	5
SIM	0.1	0.32	0.27	0.13	0.85	0.08	0.72
AN	0.1	0.28	0.31	0.16	0.73	0.09	0.95
SIM	0.25	0.27	0.23	0.11	0.71	0.06	0.69
AN	0.25	0.26	0.30	0.15	0.69	0.08	0.91
SIM	0.5	0.19	0.17	0.10	0.50	0.05	0.60
AN	0.5	0.19	0.21	0.11	0.50	0.06	0.66

* SIM - simulation, AN - analytic

The results presented in Tables III, IV, V and VI show that the locking delays may cause significant degradation of system performance. This degradation depends heavily on the number of disks I/O operations done while keeping a file locked (i.e., on the probability of delay). Also, when one file is shared by transactions of several types, then the degradation of performance is higher than in the no-sharing case; especially for large numbers of transactions. The accuracy of results appears to be acceptable. The cost of the analytic solution is from 100 to 200 times less than the cost of simulation solution depending on the values of the model parameters, especially probability of delay.

In the second example, the performance measures are computed for a distributed system containing two identical hosts. Each host includes one file disk with three files on it. Performance measures were calculated for the analytic and the approximate models in the no-sharing case (as shown in Fig. 6), and when one file is shared (as shown in Fig. 7). A simulation model was not run since, as shown in Tables III, IV, V and VI, the delays can be modeled analytically with reasonable accuracy. The comparison of CPU time spent to solve each model using the RESQ2 package showed that the approximate solution needs about 70 percent of CPU time of the analytic non-approximate solution. When the number of transactions, files and disks increases, then the percentage of the CPU time spent on approximate solution versus the analytic one, decreases. The results of the comparison of performance measures are presented in Tables VII and VIII.

Table VII. Performance measures for the distributed system model when no file is shared by transaction types.

Model*	Host Number	Transaction Types	Number of Transactions of Each Class	Probability of Delay	Performance Measures			
					Utilization		Throughput	Response Time
					CPU	Disk	[1/sec]	[sec]
AN	1	1; 2; 5	2; 2; 2	0.1	0.38	0.22	0.59	7.76
	2	3; 4; 5	2; 2; 2	0.1	0.99	0.45	1.18	3.24
AP	1	1; 2; 5	2; 2; 2	0.1	0.39	0.24	0.62	9.61
	2	3; 4; 5	2; 2; 2	0.1	0.99	0.48	1.24	4.84
AN	1	1; 2; 5	2; 2; 2	0.9	0.37	0.22	0.58	7.9
	2	3; 4; 5	2; 2; 2	0.9	0.98	0.44	1.17	3.3
AP	1	1; 2; 5	2; 2; 2	0.9	0.39	0.24	0.61	9.61
	2	3; 4; 5	2; 2; 2	0.9	0.99	0.48	1.24	4.84

* AN - analytic, AP - analytic approximate

The results presented in Table VII show that, when the CPU is the bottleneck in the system, then changing the probability of delay does not have any significant impact on the system performance.

Table VIII. Performance measures for the distributed system model when one file is shared by transaction types.

Model*	Host Number	Transaction Types	Number of Transactions of Each Class	Probability of Delay	Performance Measures			
					Utilization		Throughput [1/sec]	Response Time [sec]
					CPU	Disk		
AN	1	1; 2; 5	2; 2; 2	0.1	0.38	0.22	0.59	7.78
	2	3; 4; 5	2; 2; 2	0.1	0.99	0.45	1.18	3.24
AP	1	1; 2; 5	2; 2; 2	0.1	0.45	0.28	0.7	8.5
	2	3; 4; 5	2; 2; 2	0.1	0.99	0.56	1.41	4.26
AN	1	1; 2; 5	2; 2; 2	0.25	0.37	0.22	0.57	8.12
	2	3; 4; 5	2; 2; 2	0.25	0.96	0.43	1.15	3.37
AP	1	1; 2; 5	2; 2; 2	0.25	0.43	0.26	0.66	9.02
	2	3; 4; 5	2; 2; 2	0.25	0.95	0.53	1.34	4.48
AN	1	1; 2; 5	2; 2; 2	0.5	0.30	0.18	0.46	10.63
	2	3; 4; 5	2; 2; 2	0.5	0.78	0.35	0.93	4.60
AP	1	1; 2; 5	2; 2; 2	0.5	0.26	0.16	0.41	14.57
	2	3; 4; 5	2; 2; 2	0.5	0.58	0.32	0.81	7.41
AN	1	1; 2; 5	2; 2; 2	0.75	0.21	0.13	0.33	15.76
	2	3; 4; 5	2; 2; 2	0.75	0.56	0.25	0.67	7.12
AP	1	1; 2; 5	2; 2; 2	0.75	0.17	0.11	0.27	22.16
	2	3; 4; 5	2; 2; 2	0.75	0.38	0.22	0.54	11.11
AN	1	1; 2; 5	2; 2; 2	0.9	0.18	0.11	0.28	19.04
	2	3; 4; 5	2; 2; 2	0.9	0.47	0.21	0.56	8.88
AP	1	1; 2; 5	2; 2; 2	0.9	0.15	0.09	0.23	26.00
	2	3; 4; 5	2; 2; 2	0.9	0.32	0.18	0.45	13.30

* AN - analytic, AP - analytic approximate

The results presented in Table VIII show that even when the CPU is the bottleneck in a system in which a file is shared by transactions of different types, then changing the probability of delay may have a significant impact on the system's performance. In this case, when the probability of delay increases, then the CPU is no longer a bottleneck in the system.

The values of the performance measures obtained are sufficiently close in most cases. For the approximate model the throughput and the response time are obtained by using as the weighting factor the ratio of the number of transactions in a host to the number of transactions in the whole system.

The comparison of the results from the models shows that the performance degradation due to locking can be significant. This degradation is higher when files are shared by different transaction types. This is why the granularity of locking is very important to system performance. For instance, if locks are on records or sectors rather than on files, then the likehood of sharing is smaller and the performance can be higher. Also the number of disk I/O operations done while keeping a file locked has a significant impact on performance.

8. Conclusion and Future Research

The approach presented in this paper permits the evaluation of queueing networks with delays due to locking of the files.

An analytic model of a distributed system permits the calculation of performance measures using hierarchical decomposition.

The non-approximate model can be solved using a queueing network analysis package (for instance, RESQ2 [9]). This solution can be applied to distributed systems with a few disks and a small number of files. However, in the case when many files are shared by transactions of different types, the locking delays are repeated and this increases the model's complexity and the cost of the solution. For instance, when there are more than 3 disks and 4 files on each disk, then this will be cheaper to solve the model using approximate solution.

It is important to notice that the model allows multiple classes of transactions and shared files to be represented, and this is crucial for performance evaluation.

The accuracy of the results, on the basis of a comparison with simulation solutions, appears to be acceptable.

There are a few problems which remain to be investigated.

The first is concerned with the granularity of locks, i.e., with whether they are on records, sectors or files. If locks are on records or sectors rather than on files, then the likehood of sharing is smaller and the performance should be expected to be higher. However, only a quantitative evaluation of the benefits can confirm this expectation.

Another problem is that of the changes due to locking in the structure of the workload executed in a system. Since the service centers (i.e., the disk and CPU server) are not entirely utilized because of transactions which locked the files and were then sent to the other service centers without releasing the lock, it is possible to execute transactions of the other types which require unlocked files. This may improve the system's performance; however, the performance (i.e., the throughput) of some types of transactions will decrease. The changing of the number of transactions of each type can minimize the locking overhead.

Finally, there is the problem of the replication of shared files. Heavily shared files can be replicated to allow more transactions to access them concurrently. However, the requirement of consistency among replicated files must be satisfied, and read and write accesses should be kept carefully distinct.

9. Acknowledgments

The author wishes to thank Domenico Ferrari for his encouragement and support during the preparation of the paper and for his helpful comments on it. She also would like to thank T. Paul Lee for helpful discussions on the subject presented in the paper, and Susan Whitford and Praful Shah for providing the data used in this paper. Thanks are also due to the Research Division of the IBM Corporation, whose queueing network analysis package RESQ2 played a very useful role in the research reported here.

10. References

[1] F. Baskett, K. M. Chandy, R. R. Muntz and F. G. Palacios, Open, Closed, and Mixed Networks of Queues with Different Classes of Customers, Journal of the ACM, 22, 2 (April 1975) 248-260.

[2] M. Blasgen, J. Gray, M. Mitoma and T. Price, The Convoy Phenomenon, ACM Operating Systems Review, 13, 2 (1979) 20-25.

[3] D. D. Clark, K. T. Pogran and D. P. Reed, An introduction to Local Area Network, Proc. IEEE, 66, 11 (November 1978) 1497-1517.

[4] D. Ferrari and T. P. Lee, Modeling File System Organizations in a Local Area Network Environment, Proc. Conference of Computer Data Engineering (April 1984); also, Report No. UCB/CSD 83/142 (October 1983) Computer Science Division (EECS) University of California, Berkeley, California 94720.

[5] D. C. Gilbert, Modeling Spin Locks with Queueing Networks, ACM Operating Systems Review, 12, 1 (1978) 29-42.

[6] A. Goldberg, G. Popek and S. Lavenberg, A Validated Distributed System Performance Model, Proc. Performance'83 (1983) 251-268.

[7] H. Kobayashi, Modeling and Analysis: An Introduction to System Performance Evaluation Methodology (Addison-Wesley, 1978).

[8] G. Popek, B. Walker, J. Chow, D. Edwards, C. Kline, G. Rudisin and G. Thiel, LOCUS: A Network Transparent High Reliability Distributed System, Proc. Eighth Symposium on Operating Systems Principles (December 1981) 169-177.

[9] C. H. Sauer, E. A. MacNair and J. F. Kurose, Computer/Communication System Modeling with the Research Queueing Package Version 2, IBM Corp. (1981).

[10] B. Walker, G. Popek, R. English, C. Kline and G. Thiel, The LOCUS Distributed Operating System, Proc. Ninth Symposium on Operating Systems Principles (October 1983) 49-70.

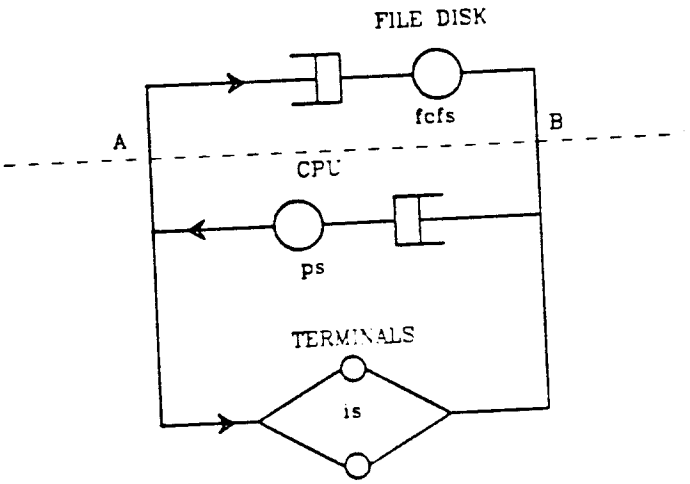

Fig. 1. The model of a system.

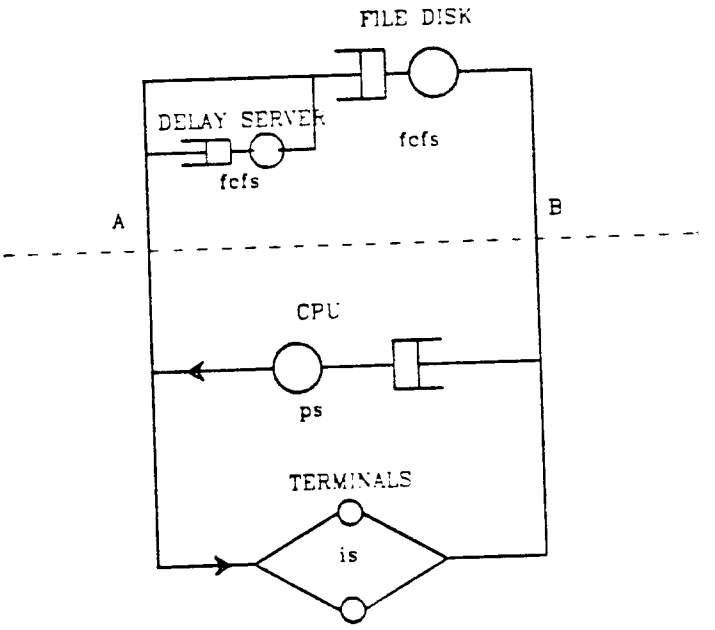

Fig. 2. The model of a system with a delay server.

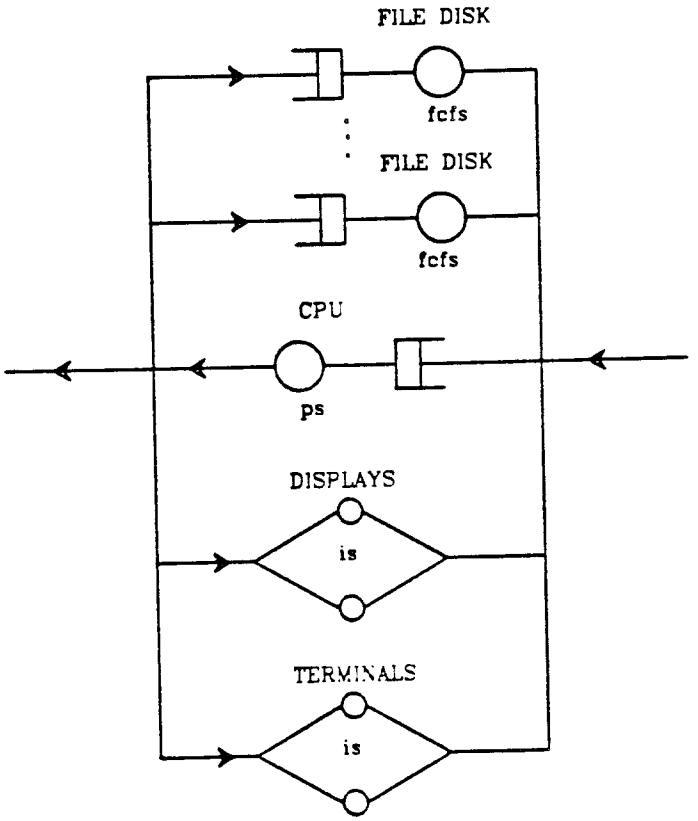

Fig. 3. The model of a host.

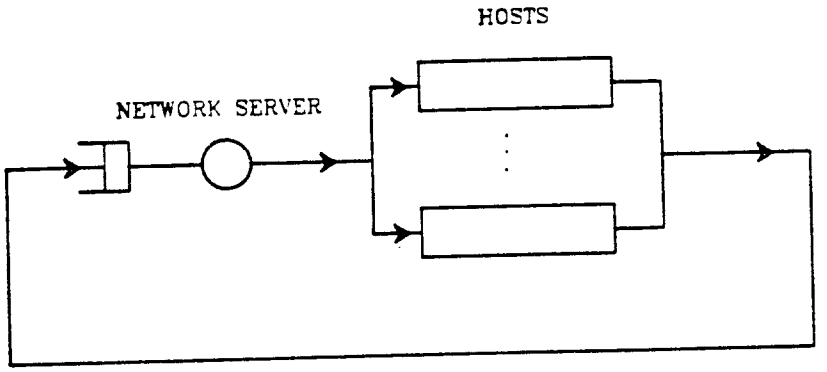

Fig. 4. The model of a distributed system.

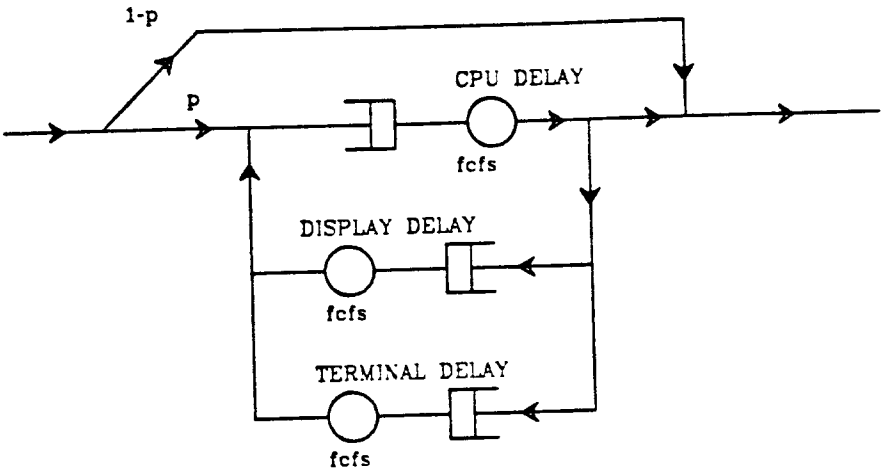

Fig. 5. The model of a single delay due to locking a file.

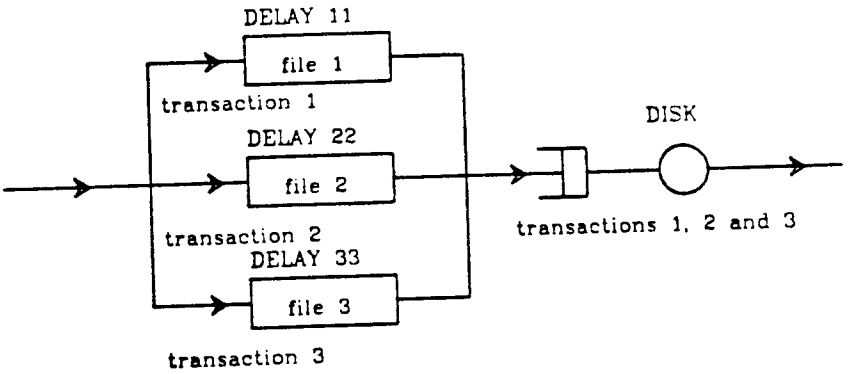

Fig. 6. The model of a file disk with delays due to accessing 3 files by three types of transactions (no file is shared by transaction types).

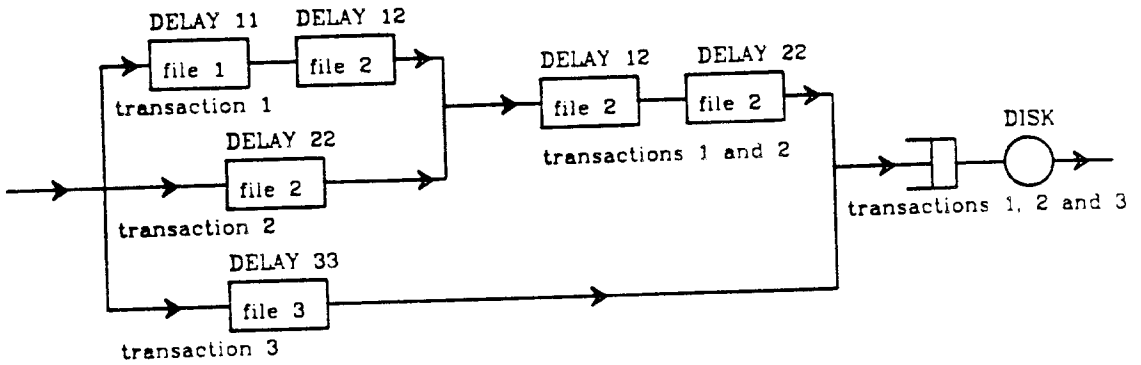

Fig. 7. The model of a file disk with delays due to accessing 3 files by three types of transactions (second file is shared by transactions of type 1 and 2)

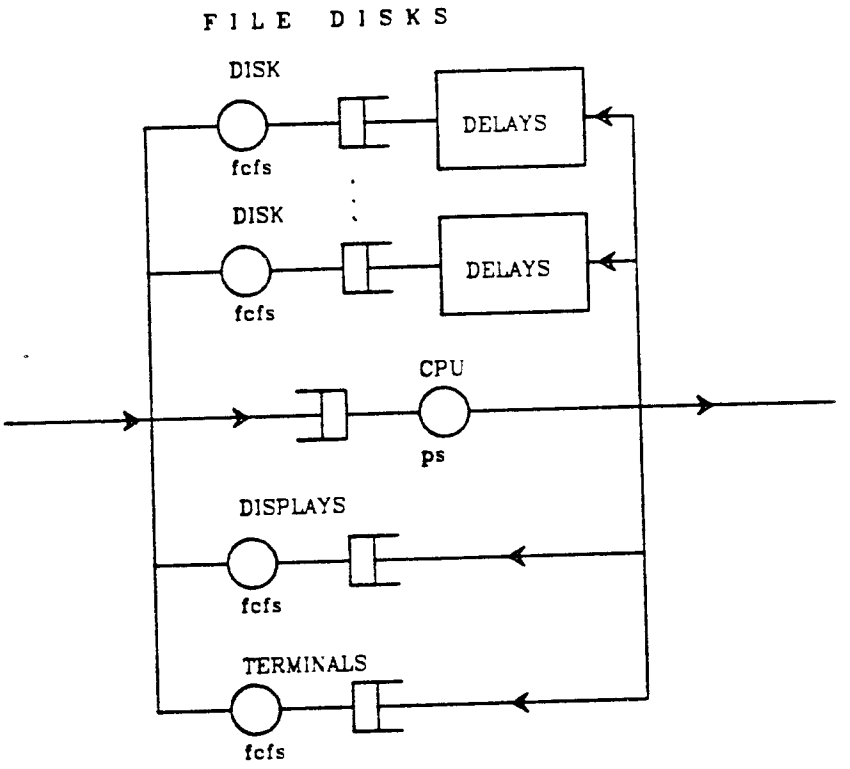

Fig. 8. An approximate model for a host.