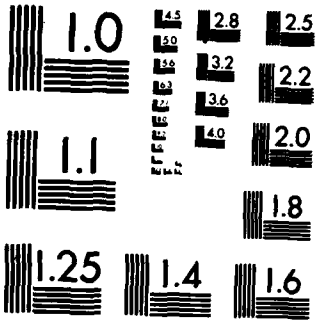
AD-A147 445 DECENTRALIZED CONTROL OF SCHEDULING IN DISTRIBUTED SYSTEMS(U) MASSACHUSETTS UNIV AMHERST DEPT OF ELECTRICAL AND COMPUTER ENGINEERING J A STANKOVIC 1/1

UNCLASSIFIED 15 MAR 84 F/G 12/1 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0

RESEARCH AND DEVELOPMENT TECHNICAL REPORT
CECOM

DECENTRALIZED CONTROL OF SCHEDULING IN DISTRIBUTED SYSTEMS

John A. Stankovic
Dept. of Electrical & Computer Engineering
University of Massachusetts
Amherst, MA 01003

Quarterly Report for the Period 16 DEC 83 to 15 MARCH 84

Distribution Statement

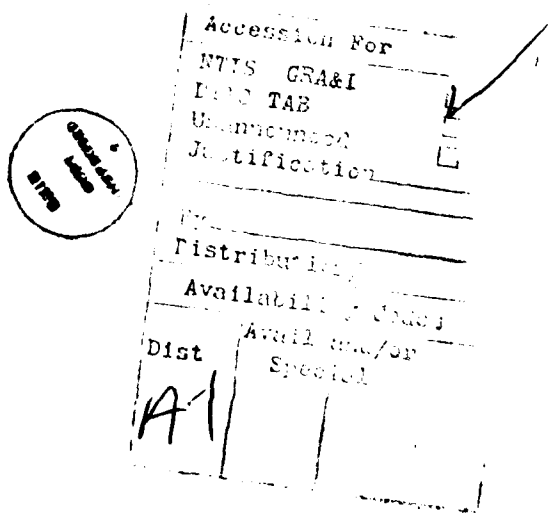
Approved for public release;
distribution unlimited

Prepared for
Center for Communications Systems (C. Graff)

CECOM

U S ARMY COMMUNICATIONS-ELECTRONICS COMMAND
FORT MONMOUTH, NEW JERSEY 07703

Table of Contents



```
--     4   OF      7
-- 1  -  AD NUMBER: A132174
--2 3  -  SEC CLASS HOLDING SYMBOL:        NPL
--                                         TRAC
--                                         NWC
--                                         TAC
-- 4  -  FIELDS AND GROUPS: 9/2  5/1
-- 5  -  PRES'D CLASSIFICATION:    UNCLASSIFIED
-- 6  -  /CORPORATE AUTHOR: MASSACHUSETTS UNIV  AMHERST  DEPT OF ELECTRICAL
--       AND COMPUTER ENGINEERING
-- 6  -  UNCLASSIFIED TITLE:       DECENTRALIZED CONTROL OF SCHEDULING IN
--       DISTRIBUTED SYSTEMS
-- 8  -  TITLE CLASSIFICATION:     UNCLASSIFIED
-- 9  -  DESCRIPTIVE NOTE:  ANNUAL REPT  16 DEC 82-15 DEC 83
--10  -  PERSONAL AUTHORS: STANKOVIC, J  A. ;
--11  -  REPORT DATE:       DEC 15, 1983
--12  -  PAGINATION:      53P
--15  -  CONTRACT NUMBER:  N00014-82-K-0315
--16  -  REPORT CLASSIFICATION:  UNCLASSIFIED
--17  -  DESCRIPTORS:       *DATA MANAGEMENT, *SCHEDULING, *MANAGEMENT
--       PLANNING AND CONTROL, *NETWORK ANALYSIS(MANAGEMENT), DISTRIBUTED
--       DATA PROCESSING, DECENTRALIZATION, CONTROL THEORY, ALGORITHMS,
--          <<P FOR NEXT PAGE>> OR <<ENTER NEXT COMMAND>>

--       DECISION THEORY, HEURISTIC METHODS, SYSTEMS ANALYSIS, BAYES THEOREM,
--       COMPUTER COMMUNICATIONS
--20  -  DESCRIPTOR CLASSIFICATION:     UNCLASSIFIED
--21  -  ABSTRACT:    THIS REPORT SUMMARIZES THE WORK AND RESULTS PRODUCED
--       DURING THIS CONTRACT OVER THE PAST YEAR.  A SUMMARY OF OUR RESULTS
--       ON THE USE OF BAYESIAN DECISION THEORY IS REPORTED. TWO MASTER'S
--       THESIS WERE COMPLETED, ONE ON BIDDING AND ONE ON STOCHASTIC
--       LEARNING AUTOMATA. WE HAVE ALSO BEEN WORKING ON AN ANALYTICAL MODEL
--       BUT LITTLE NEW PROGRESS OVER THAT REPORTED IN THE LAST QUARTERLY
--       REPORT HAS BEEN MADE IN THIS AREA. PROGRESS WAS MADE ON A
--       DECENTRALIZED SCHEDULING APPROACH FOR SYSTEMS WITH TASKS HAVING
--       REAL-TIME CONSTRAINTS. THIS REPORT ALSO BRIEFLY STATES OUR OVERALL
--       CONCLUSIONS AND MAKES SEVERAL RECOMMENDATIONS FOR THE WORK TO BE
--       DONE IN THE FINAL YEAR OF THIS CONTRACT. SECTION 5 LISTS THE PAPERS,
--       THESIS, AND PRESENTATIONS PRODUCED THIS YEAR. SECTION 6 CONTAINS
--       THE REQUIRED BUDGET INFORMATION, AND FINALLY THE REPORT ENDS WITH A
--       COPY OF THE DISTRIBUTION LIST. (AUTHOR)
--22  -  ABSTRACT CLASSIFICATION:       UNCLASSIFIED
--23  -  INITIAL INVENTORY:        2
--24  -  LIMITATION CODES:   1
--25  -  SOURCE CODE:        0
--26  -  SOURCE CODE:        AD55800
--          <<P FOR NEXT PAGE>> OR <<ENTER NEXT COMMAND>>

--27  -  DOC MEMT LOCATION:          NTIS
--40  -  GEOPOLITICAL CODE:        8501
--41  -  TYPE CODE:    0
```

1.0 Introduction

This report summarizes the work performed during the ninth quarter of this contract, from December 15, 1983 to March 14, 1984. The work in this quarter concentrated on three areas: scheduling algorithms based on bidding, scheduling algorithms for systems with hard real-time constraints and on scheduling based on stochastic learning automata. Section 2 of this report describes the progress made in these areas.

We also note that the paper entitled "Bayesian Decision Theory and It's Application to Decentralized Control of Job Scheduling" produced earlier in this contract has been accepted for publication in IEEE Transactions on Computers.

2.0 Main Text

2.1 Bidding

In this quarter we concentrated on designing extensions to our simulation program which implements our decentralized bidding algorithm (described in detail in previous reports). As a reminder, the main idea behind the algorithm is to treat all the information needed to make a scheduling decision as input to a McCulloch Pitts neuron. If the excitations of the neuron surpass a threshold then a task is not considered as a candidate for bidding, else it is. The extensions worked on in this quarter involve two areas. The first is modifying the model of a process from simply one object with size χ bytes to considering the a process to be collection of objects each of different size. Further, these objects of a process are modeled to have certain characteristics including whether they are unshared, shared serially or shared simultaneously. The second extension involved developing formulas for inclusion in our McCullock Pitts neuron. The approach will be to incrementally include these formulas into our simulation model. We now present the formulas.

Factors for the McCulloch Pitts Neuron

Individually the factors described here are simple, but taken together they provide for a very sophisticated scheduler. A major research goal is to normalize the weights, W, of the various factors, discover dependencies among the weights and adapt them as a function of system-wide workload.

1. Queue Length Factor = QL

   Consider

   Q_W = length of the queue of jobs waiting to become active (e.g., because the level of multiprogramming has been reached at this host)

   Q_A = The number of active tasks at this site

Then

$$QL = W_Q \left(\frac{1}{Q_W + Q_A + 1}\right).$$

Note: Queue lengths for certain resources are used in other factors (see the segments shared serially factor and position in wait queue factor).

2. Memory Factor = M.

Consider

M_{AVAIL} = amount of free memory at this host

$M_{current}$ = amount of memory assigned to this task

M_{new} = new memory requirement of this task

Then

$$M = W_M \left[\frac{(M_{AVAIL} - (M_{new} - M_{current}))}{M_{AVAIL}}\right]$$

Note: The memory requirement of a task at time t is the sum of its segment sizes. The fact that some of its segments may be simultaneously shareable is used in the next factor.

3. Simultaneously Shared Segments Factor = Si.

Let N_{Si} = the number of tasks sharing this segment of task P

Initially we consider two forms:

$$W_{Si} \cdot N_{Si}$$

or

$$W'_{Si}\, e^{N_{Si}}$$

Then $Si = \sum_{\sigma} W_{Si}\, N_{Si}$ or $\sum_{\sigma} W'_{Si}\, e^{N_{Si}}$

where σ = the set of this task's shared simultaneously segments.

4. Segments Shared Serially Factor = Se.

Let N_{Se} = the number of tasks waiting for this segment at this site

$$S_E = \sum_{\sigma} W_{Se} N_{Se}$$

where

σ = the set of this task's shared serially segments.

5. Distributed Group Factor = DG.

Let N_{DG} = the number of tasks of this distributed group at this host

We have two forms for this factor:

$$DG = \delta W_{DG} \left(\frac{1}{N_{DG} + 1}\right)$$

or

$$DG = \delta W'_{DG} \, e^{(1-N_{DG})}.$$

where if $N_{DG} = 0$ then $\delta = 0$, else $\delta = 1$.

6. Cluster Factor = C.

Assuming that this task is a member of a cluster, then Let N_c = total number of tasks in this cluster, $N_C \geq 1$.

N_{cs} = total number of tasks of this cluster at this site,
$N_{CS} \geq 1$.

Two forms of C are:

$$C = W_c \left(\frac{N_c}{N_{cs}}\right)$$

or

$$C = W_c' e^{\frac{-\alpha(N_c - N_{cs})}{N_c}}$$

where α is a normalizing constant.

7. Priority Factor = Pr.

Let the priority of this task be P_p.

Then $Pr = W_{Pr} P_p$.

8. Precedence Factor = P.

At this time we consider precedence constraints between tasks of the form, task A must precede task B (or equivalently task B must not begin until task A completes). We also assume that task B would in general, use some of the output produced by A. Therefore, if task B were at the same host as task A it would be beneficial for it to remain there as far as obtaining (some of) its input was concerned. In general, a task can have any number of preceeding tasks.

Let O = the sum total (size) of the output produced at this site for this task.

Then we can consider the incentive for keeping this task here to be either:

$$W_p O$$

or

$$W_p' e^{O}.$$

On the other hand, if task B were at another site then the output produced by A would be subject to a communication cost providing incentive to move task B. We represent this incentive by

$$-W_p'' O' C_O$$

or

$$-[W_p''' e^{O'} C_O]\delta$$

where O' = total amount of output (size) produced at other sites for this task,

C_O = estimated cost per unit size of moving the output to this host, and

$\delta = 0$ if O' is zero, otherwise $\delta = 1$.

Then $P = W_P O - W_P'' O' C_O$

or

$$P = W_P' e^{O} - [W_P''' e^{O'} C_O]\delta.$$

9. CPU time requirement factor = T.

Let T_U = time used so far

T_R = time required (if known).

We have two forms:

$$T = W_{CPU}(\frac{T_U}{T_R})$$

or

$$T = W'_{CPU}\, e^{(T_U/T_R)}.$$

10. Position in Wait Queue Factor = W_t.

Let W_Q = position in wait queue of resource j where j is a type of resource that is available elsewhere (e.g., printers and other I/O devices).

Then

$$W_t = W_{Wt}\, W_Q$$

or

$$W_t = W'_{Wt}\, e^{W_Q}.$$

11. Inhibitors

I_1 = this task moved x times already, so $I_1 = 0$, else 1.

I_2 = this task needs a resource that is fixed at this site, so
I_2 = 0, else 1

Then the McCulloch Pitts Neuron (MPN) in its final form could be:

$$MPN = \prod_{i=1,2} I_i \left(\sum_j W_j \cdot Factor_j\right)$$

or

$$MPN = I_1 \cdot I_2 \cdot [QL + M + Si + Se + DG + C + Pr + P + T + Wt]$$

In this formula DG, C and P use network-wide information. Another aspect of our research is to extend some of the other factors to deal with network-wide information such as M and T as well as including additional factors such as the number of outstanding bids.

2.2 Real Time Constraints

This research involves scheduling periodic and nonperiodic tasks, with real-time constraints, on processors in a network. A nonperiodic task is one that occurs in the system just once and at unpredictable times. Upon arrival, it is characterized by its deadline and its computation time. No priorities are assigned to tasks other than those induced by deadline. It is assumed that there is no advantage to completing a task before its deadline. Thus response times are not crucial, deadlines of course are.

In this quarter we have been considering extensions to this work where tasks can conflict with other tasks over resources (besides the CPU). We have developed a heuristic for dealing with resource constraints and are in the process of evaluating the heuristic.

In the development of the heuristic we make the following assumptions:

1. When a task arrives, its resource requirements are known.

2. A task also indicates whether it requires the CPU, some other device, such as an I/O device, or both.

3. Tasks that require CPU or devices, but not both, can execute concurrently.

Suppose a set of guaranteed tasks exist on a node and a new task arrives. Our approach consideres that the new task can be guaranteed only if it does not cause previously guaranteed tasks to miss their deadlines. Because of the overwhelming computations needed to perform such a check exhaustively, we need a heuristic approach which can be summarized as follows: The following heuristic functionis applied to each task in the system, including the newly arrived task.

$$H(t) = K1*X1(t) + K2*X2(t) + K3*X3(t)$$

where t is a task, K1, K2 and K3 are adaptive coefficients and

1. X1(t) takes into account the resource and device requirements of task t as well as the resource and device utilization,

2. X2(t) takes into account the laxity of task t, and

3. X3(t) takes into account the computation time of task t.

We do not describe the components of the above formula in detail, because they still are under development, but it is important to note that these factors take all the important issues into account.

From the result of the function application to each task, the task with the smallest value is chosen as a candidate to be scheduled first. Assuming that the task would complete execution, i.e., releases resources and devices, the heuristic function is applied again to each of the remaining tasks to determine the next task to be scheduled, and so on. When a task is chosen to be scheduled next by using function H(t), our algorithm also checks that it is guaranteed to meet its requirements. If this fails for any of the tasks in the system, it means that our heuristics did not find a way to guarantee all tasks in the system. As a result, the newly arriving task is not guaranteed and, of course, all previously guaranteed tasks are not affected.

The performance of this technique is intimately related to the ability of the heuristic function to find a good schedule, assuming one exists. Obviously, the choice of values for the coefficients will affect the performance of the heuristic. The results of the tests performed so far, on a set of task samples by using a properly tuned set of coefficients, are encouraging. We have observed that the heuristic guarantees around 80% of the tasks, i.e., when a set of tasks arrive and are known (by exhaustive search) to be guaranteeable, the heuristic technique just described guarantees the tasks in 80 out of 100 instances. To increase the percentage of success on a set of tasks, we propose to adapt the coefficients of the function using the information derived from applying the heuristic to the given set of tasks and reapply the function to the same set of tasks. We term this technique, by which success on a specific set of tasks can be increased, as local adaptation.

Since our algorithm is designed to function in a dynamic environment, the K1, K2 and K3 values will also have to adapt to changing system loads. We propose to continue our study of the heuristic technique in the coming months.

2.3 Stochastic Learning Automata

This quarter was spent evaluating simulation results produced as part of a masters thesis, as well as improving the writing of one thesis itself. The results produced so far indicate that additional simulations are required for statistical accuracy. This should be accomplished within the next month at which time the thesis can be defended. Details about the results themselves should be possible in the next quarterly report.

3.0 Conclusions and Recommendations

In general we are moving into more and more sophisticated scheduling algorithms. Simulation studies, to date, have been very encouraging. We hope to continue our progress in understanding decentralized scheduling and decentralized control in general. We believe that the simulations to be performed in the future will show a level of sophistication not yet seen in the literature. This should continue to produce significant new results.

4.0 Budget

Spent prior to this report: 66,464.62

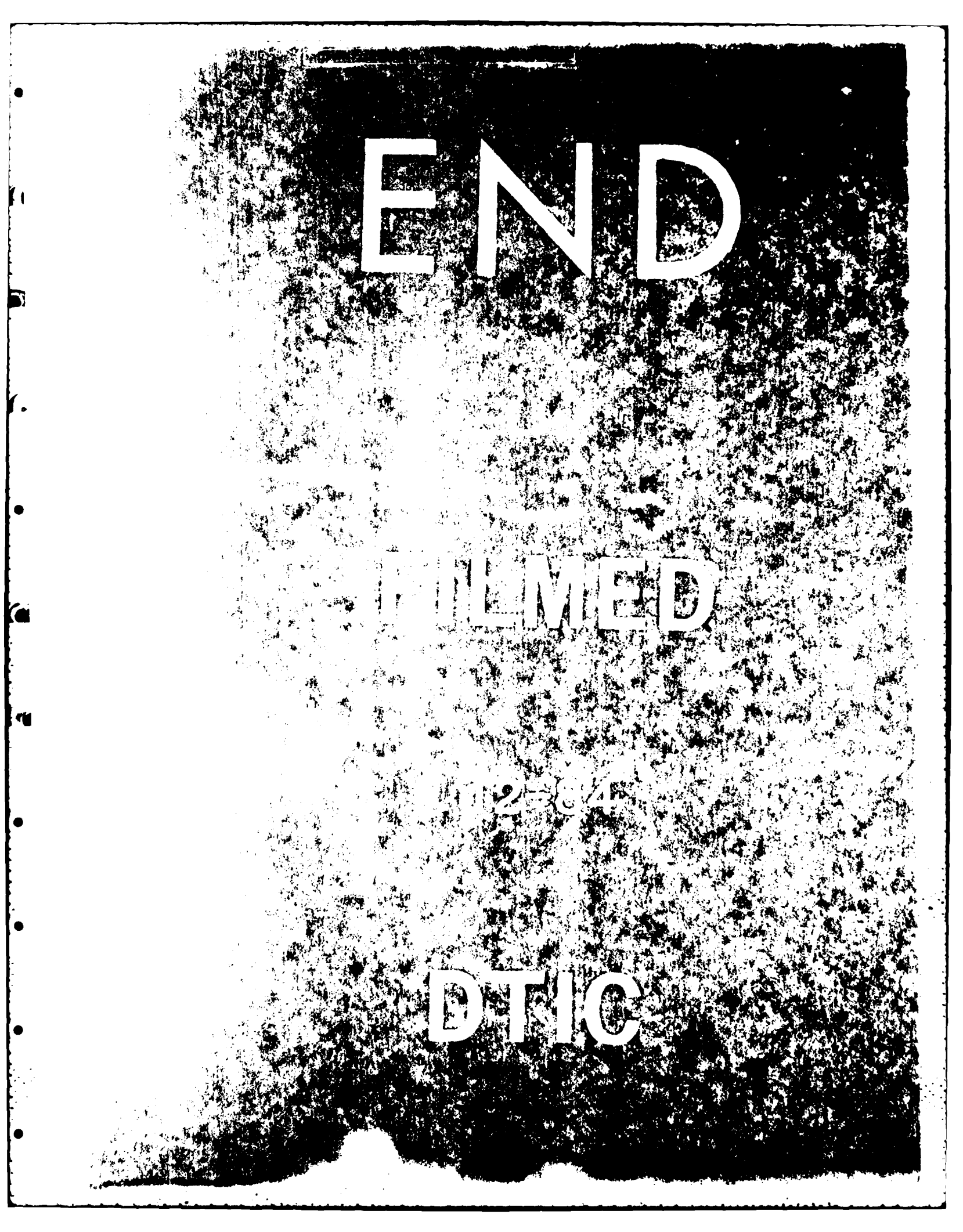
Current Budget for Third Year of Contract

	Planned	Actually Spent
December (83)	4,000	3,319.21
January (84	4,000	
February	4,000	
March	4,000	
April	4,000	
May	4,000	
June	7,000	
July	7,000	
August	7,000	
September	5,000	
October	5,000	
November	5,000	
Decemeber	4,3665.38	

Total: $130,830.00 (Planned and Spent)

5.0 Distribution List

Defense Technical Information Center Cameron Station Alexandria, VA 22314	2 copies
Commander USAERADCOM ATTN: DELSD-L-S Fort Monmouth, NJ 07703	2 copies
Commander USACECOM ATTN: DRSEL-COM-RF (Dr. Klein) Fort Monmouth, NJ 07703	1 copy
Commander USACECOM ATTN: DRSEL-COM-D (E. Famolari) Fort Monmouth, NJ 07703	1 copy
Commander USACECOM ATTN: DRSEL-COM-RF-2 (C. Graff) Fort Monmouth, NJ 07703	10 copies


END
FILMED
DTIC