A Sensitivity Study of the Clustering Approach to Workload Modeling *

Maria Calzarossa† *and Domenico Ferrari*

Computer Science Division

Department of Electrical Engineering and Computer Sciences

and the Electronics Research Laboratory

University of California, Berkeley

ABSTRACT

If an adequate separable model of an interactive system can be built, if the users' behaviors can be accurately modeled by probabilistic graphs, and if the workload and the model of it to be constructed are stationary, then a perfectly accurate workload model of a given workload can be constructed by grouping together all commands with *identical* characteristics. If a clustering technique which groups together commands with *similar* characteristics is applied, is the workload model produced still acceptably accurate? This paper addresses this question (and other similar ones) by taking an experimental approach. The answer is that, in the case considered here, the accuracy is still acceptable if certain conditions are satisfied.

* The research reported here was supported in part by the Consiglio Nazionale delle Ricerche, Rome, Italy, by the University of California under a MICRO Program grant, and by the NCR Corporation.

† Author's permanent affiliation: Istituto di Analisi Numerica, Consiglio Nazionale delle Ricerche, University of Pavia, Pavia, Italy.

1. Introduction

In a paper published in 1984 [Ferr84], the validity of applying clustering techniques to the design of an executable model for an interactive workload was discussed. The following assumptions, intended not to be necessarily realistic but to provide sufficient conditions for the applicability of clustering techniques, were made:

(1) The system whose workload is to be modeled is an interactive system, and its performance can be accurately evaluated by solving a separable closed queueing network model.

(2) The behavior of each interactive user can be adequately modeled by a probabilistic graph (called a *user behavior graph*); in such a graph, each node represents an interactive command type, and the duration of a user's stay in the node probabilistically equals the time the user spends typing in a command of that type, waiting for the system's response, and thinking about what command should be input next.

(3) The interactive workload to be modeled is stationary, and the workload model to be constructed is intended to reproduce its global characteristics (not those of some brief excerpt from it exhibiting peculiar dynamics), hence to be stationary as well.

It was shown in [Ferr84] that, under these assumptions, clustering command types having the same probabilistic resource demands does not affect the values of the performance indices the evaluators are usually interested in, provided the visit ratio to each node in the reduced (i.e., post-clustering) user behavior graph is equal to the sum of the visit ratios the cluster's components had in the original graph.

Since the reduction we have just described is equivalent to replacing each cluster with one or more representatives of its components, and since this is also the goal of applying clustering techniques to the construction of executable workload models substantially more compact than the original workload to be modeled, this result shows that such techniques are valid (i.e., produce accurate models) when the assumptions and the conditions mentioned above are satisfied.

One condition which in practice is never satisfied, however, is that the clustered commands are characterized by exactly the same resource demands. In fact, clustering algorithms are nontrivial just because they have to recognize "nearness" among commands with different characteristics, and group those and only those commands whose resource demands are sufficiently similar (where the notion of *similarity* is to be defined by introducing that of *distance* between two commands). Thus, the question of the sensitivity of a workload model's accuracy to the inevitable dispersion of the characteristics of a cluster's components immediately arises. This paper reports on an experimental investigation intended to provide an answer to this question.

The next section, Section 2, discusses the approach that was taken and the rationale for it. In Section 3, we describe the measurement and the modeling of a real workload to be used as the reference workload in our study. The design of the experiment is detailed in Section 4. The experiment and its results are then summarized in Section 5, and conclusions are given in Section 6.

2. The Approach

The sensitivity problem alluded to in the previous section can be more precisely formulated as follows. We know that, if an adequate separable model of an interactive system can be built, if the users' behaviors can be accurately modeled by probabilistic graphs, and if the workload and the model of it to be constructed are stationary, then a workload model in which all commands with identical characteristics are grouped together and modeled by a single representative is an accurate model of the given workload (i.e., the model produces the same values of the performance indices of interest as the modeled workload when it is processed by a given system). This is true, of course, provided the visit ratios of the workload model's components equal the sums of those of the corresponding workload components. If we now apply a clustering algorithm to the given workload, thereby obtaining clusters of similar, but not identical, commands, and we build a workload model by assembling cluster representatives (usually one per cluster, for instance with

demands corresponding to those of the cluster's center of mass), by how much will the values of the performance indices produced by the workload model running on the given system differ from those produced by the workload to be modeled?

As with several other problems, this could be attacked by a mathematical approach or by an experimental one. While a successful mathematical analysis of the sensitivity of the major indices to the dispersion in the resource demands of the commands being clustered together would provide more general results, it would also be likely to require the introduction of simplifying assumptions (for example, having to do with the distributions of the resource demands in a cluster around its center of mass) whose validity would be neither self-evident nor easy to verify experimentally.

On the other hand, an experimental approach achieves results which, strictly speaking, are only applicable to the cases considered in the experiments. Extrapolations to other systems, other workloads, other environments usually require faith, along with experience, common sense, and familiarity with real systems and workloads. This inherent lack of generality is somehow counterbalanced, however, by the higher degree of realism that is achievable with an experimental investigation. In particular, when in a study the properties of workloads are to play a crucial role (there are very few studies indeed in which this is not the case!), using a mathematical approach is bound to raise about such properties questions that are either very difficult or impossible to answer. Primarily for this reason, and knowing very well the limitations in the applicability of the results we would obtain, we decided to adopt an experimental approach.

Since the question we were confronted with had never been answered before (nor, to our knowledge, had it been asked), we felt that our choice was justified by the exploratory nature of the study. If the resulting sensitivity were to turn out to be high, we could conclude that not even under the above assumptions can clustering techniques be trusted to provide reasonable accuracy in all cases and hence should not be used, or used with caution in those cases (if they exist) in which their accuracy might be acceptable. If, on the other hand, the sensitivity were low, then we could say that, in at least one practical case, clustering techniques would have been shown to work adequately (of course, under all the other assumptions listed above).

The rationale of this investigation might be questioned by asking why it would not be more convenient to test the validity of clustering techniques directly, that is, by comparing the performance indices produced by a real workload to those produced by an executable model built according to a clustering technique. Our answer, which explains also some of the aspects of our experimental design to be described in Section 4, is that, in this study as well as in [Ferr84], we are more interested in understanding the limitations and the implications of clustering and other workload model design methods than in evaluating the accuracy of clustering in a particular case. In other words, we are not so much keen on finding out whether the errors due to clustering are of the order of 10% or of 80%, but we want to be able to understand why they are only 10% or as large as 80%, respectively. Thus, we need to decompose the total error into the contributions to it of the various discrepancies that any real situation exhibits with respect to the ideal one. This paper describes a study primarily performed to assess the magnitude of one such contribution, that of the dispersion of the resource demands of clustered commands.

An experimental approach, in the case being considered here, requires first of all that a workload for the experiment be selected. Then, that workload is to be measured, in order to obtain the values of the parameters defined by the desired characterization. These first two steps are described in Section 3.

Next, an executable workload model is to be built by applying a clustering technique (see for example [Agra76], [Arti78], [Ferr83]) to the real workload selected. Then, the workload and its model are to be run on the same system, so that the model's accuracy can be evaluated by comparing the performance indices produced by them. As our study is to try to isolate the sensitivity of that accuracy to the differences in demands among the commands that have been grouped into the same cluster, these differences must be made the only source of inaccuracies in the performance produced by the model. To isolate this contribution to the error from all of the others, the latter sources should be eliminated. Section 4 includes details about how this result was achieved

in our study, as well as about other important aspects of the design of our experiment.

Finally, the experiment is to be carried out, and its results interpreted. These last steps are described in Sections 5 and 6 of this paper.

3. The Measurement and Modeling of a Workload

This section describes the workload measurement experiments we performed. The first step was the selection of a real workload to be used as the reference workload. Among the various possibilities we had, we analyzed a one-day workload collected on a VAX 11/780 in the Computer Science Division at the University of California at Berkeley, running under the Berkeley UNIX† operating system.

The measured workload was generated by 60 different users on October 1, 1984. The data were extracted from the standard UNIX accounting files. Since we considered only the command types as the basic components of the workload, an initial reduction phase was necessary. All the commands were first grouped into command types. In our workload, we identified 113 different command types. The most popular commands were: *ls* (entered 404 times), *mail* (379), and *vi* (273). Figure 1 shows the cumulative frequency distribution of all the 113 command types. It is interesting to observe that 41 different command types accounted for 90% of the total number of commands in the workload.

To characterize each command in terms of physical resource demands, we chose the three most significant variables: the CPU time consumed (i.e., user time plus system time), the memory space required, and the number of disk I/O blocks transferred. In characterizing each command type, the first two parameters were described by their mean values and their coefficients of variation, whereas the third was only described by its mean value. The coefficient of variation of the number of disk blocks transferred was not included since that number was used to compute the mean CPU burst for each command, and its variability was therefore already reflected in that of the number of visits to the CPU. On that machine and that day, we found that only three command types, i.e., *troff, vi,* and *mail,* used more than 45% of the total CPU time. Furthermore, only two command types (*mail* and *vi*) required more than 50% of the global memory space.

Each command type is represented by a node in the user behavior graph G. To find "similarities" among command types, and to build classes of command types having homogeneous characteristics, we applied a clustering technique (see for example [Ande73]).

The *k-means* nonhierarchical clustering algorithm [Hart75] was adopted. The command types were considered as points in the m-dimensional space $\mathbf{R}^m$, m being the number of parameters used to characterize each component.

The method starts at the beginning with one cluster containing all the data and then it subdivides the points into k classes, k=1,2,..., until an optimal partition is obtained. At each iteration, the center of mass of each cluster is computed. The algorithm minimizes the sum of squares of the Euclidean distance in $\mathbf{R}^m$ between each component and the center of mass of its class.

However, since the number of iterations could be very high, in practice one chooses a local optimum configuration, preassigning the value of k and using an indicator of the goodness of a partition. As such an indicator, Hartigan [Hart75] suggests the overall mean square ratio, which measures the reduction of within-class variance between partitions into k and k+ 1 classes, respectively.

In our case, we applied the *k-means* algorithm to 40 points, i.e., the first 39 heaviest commands plus a "catch-all" command grouping together all the 74 remaining ones. The rationale for this choice will be explained in the next section.

Each command type is described by a 5-tuple of parameters, the means and the coefficients of variation of the variables introduced above.

---

† UNIX is a trademark of AT&T Bell Laboratories.

The application of the clustering technique yields an optimal subdivision of the workload into 8 clusters. The coordinates of the centers of mass of the parameters of each cluster are presented in Table 1. Thus, our resource-oriented characterization identified 8 clusters of command types whose resource demands correspond to the clusters' centers of mass. These clusters were considered as the nodes of the reduced user behavior graph G' that was obtained from G.

G and G' were characterized by two matrices , $\mathbf{P}=[p_{rs}]$ and $\mathbf{P}'=[p'_{uv}]$, respectively, which were computed directly from the real workload. p_{rs} and p'_{uv} represent the probabilities of visiting class s (s=1,2,...,40), and v (v=1,2,...,8), respectively, coming from class r (r=1,2,...,40), and u (u=1,2,...,8), respectively. Since the change from G to G' does not modify the visit ratios of the command types, the results in [Ferr84] still hold.

Our main goal was to study the causes which make G' not a perfectly accurate model of G. The design of the experiment intended to validate this clustering-based workload model will be presented in the next section.

4. The Design of the Experiment

This section describes the objectives, the factors, the context, and the setup of our experiment.

The primary goal of the investigation was, as mentioned several times in the preceding sections, to evaluate the sensitivity of the accuracy of a clustering-based workload model to the dispersion of the resource demands of the clustered commands. Other, secondary but interesting, questions that arose during the study called for the addition of a few more goals to the primary one. These questions, and the corresponding secondary goals of our experiment, will now be described in some detail.

In the previous section, the three variables chosen to characterize the resource demands of each command have been introduced. It has also been stated that commands were first grouped into command types, and that each type was characterized by five variables: the means of the distributions of the three characterizing variables for each command of that type plus the coefficients of variation of two of those distributions (CPU time and memory space). The question about how sensitive the workload model's accuracy is to the variabilities of demand distributions is a natural one to ask.

Another natural question, stemming directly from the result that motivated this study, is the one about the influence of the separability assumption on the workload model's accuracy. Of course, an experimental investigation to answer this question requires the use of a separable queueing network model along with that of either a real system or a non-separable model.

Finally, the question about the relationships, if any, between workload model accuracy (which has in our study a performance-oriented definition) and system bottlenecks (which severely influence performance) came up quite expectedly during the experimental design phase of the study, and suggested yet another goal to be pursued.

The factors of the experiment were chosen as dictated by the goals described above:

A. The dispersion of the command type resource demands around each cluster's center of mass.

B. The variability of resource demand distributions around the mean resource demands of each command type.

C. The separability of the system or system model by which the workload to be modeled and its model are processed.

D. The existence, location, and extent of bottlenecks in the system that has processed or will process the given workload and its model.

Having selected the factors of the experiment as well as the workload to be used in it, a decision remained to be made about the system on which the experiment would be performed.

The alternatives were a real system (in particular, the one whose one-day workload had been measured) and a modeled one. As usual, each solution had complementary advantages and disadvantages, but the one based on a real system would have made it impossible or extremely difficult to include resource demand distributions, separability, and bottlenecks among the factors. Thus, a simulated system was chosen as the experiment's testbed, as an analytically modeled one would have imposed excessive limitations on the model, given that the use of IBM's RESQ2 package [Saue82] [Lave83] had been decided as soon as a model-based solution was selected. The system's model we chose was as simple as needed to be compatible with the requirements of the study, mainly because of the large number of simulation runs that were expected. A diagram of the queueing network that was simulated is shown in Fig. 2.

In the model, all users are statistically identical, in that they obey the same user behavior graph. Each node in G or G' is represented by a class, and processes are assigned to a class as soon as they exit from their terminal. Once a process has been assigned to a class, memory is allocated to it. The process then cycles through the CPU and the two I/O devices until it terminates, at which time it releases its memory space. The values of service times in the CPU, memory space demands, and branching probabilities were easily obtained from the values of the node's characterizing variables, whereas the I/O service times were based on measurements performed independently of those reported on in Section 3, and terminal times were set so as to match the value of mean CPU utilization measured during the two consecutive hours of highest activity on the day the workload was measured. The use of a mean value of terminal time differing from the one that was part of the characterization of the measured workload during that two-hour period was required because of the decision to keep constant throughout the simulation the number of active terminals.

The levels of the factors were the following:

A. As described in Section 3, the workload model consisted of 8 classes, derived from the 40 original ones by applying to them the *k-means* clustering algorithm,, and assigning to each of these 8 classes the resource demands corresponding to the coordinates of the cluster's center of mass. Thus, the two levels of this factor can be labeled "40 classes" and "8 classes", respectively.

B. Besides the measured distributions, the resource demands of the commands of each type were assumed to have a deterministic (DET), hypoexponential (HPO), exponential (EXP), or hyperexponential (HPR) distribution. Thus, this factor had 5 levels, corresponding to the following values of the coefficient of variation: the measured ones, 0, 0.5, 1, and 1.5.

C. The queueing network in Fig. 2 is not separable because of the presence of a passive resource (the memory). Eliminating the memory makes it separable, since the CPU server is a processor-sharing one, and the I/O servers are exponential with an FCFS discipline. Thus, this factor's two levels are: model with memory, and model without memory.

D. Because of the decision to try to reproduce the two hours of highest activity at the CPU, the system and hence its model are CPU-bound. Thus, the levels of this factor correspond to different CPU utilizations, that is, in our model, to different values of mean terminal time. We chose, along with the nominal 40 seconds for this value, a higher (60 seconds) and a lower one (5 seconds).

As will be apparent in the next section, where the main results of the experiment are presented, we did not feel that the objectives of the study would justify a full factorial or even fractional factorial design. Thus, a two-factors-at-a-time approach was deemed sufficient, one of the two factors being, of course, A, the one which allows the workload model's accuracy to be evaluated.

5. The Experiment and Its Results

The system model we used in our experiment was based on simulation. It is known that one of the major problems of simulation is the statistical variability of the output. This variability

makes it necessary to estimate the accuracy of the simulation results, for instance by producing confidence interval estimates.

To obtain these estimates, we chose, among the possibilities provided by the RESQ2 package, the independent replications method. The number of replications for each run was set to 10, and the confidence level to 90%.

In the first runs, we considered the global user behavior graph G consisting of 40 different commands and the reduced one, G', consisting of 8 classes. The objective of these runs was to evaluate the sensitivity of the accuracy of the workload model to the dispersion of the resource demands obliterated by the use of a clustering technique.

Table 2 shows some of the values of the performance indices obtained by the G and G' simulations. The relative errors between the values v obtained from G and v' obtained from G' were computed in the following way:

$$e = \frac{|v - v'|}{v}$$

In our case, all the relative errors for these indices (G versus G') were lower than 15%. The most important performance indices we were interested in were the mean terminal response time R, the mean terminal throughput X, and the mean CPU utilization U. The errors for these indices were 3.66%, 2.68%, and .88%, respectively.

The second set of simulation runs was dedicated to the testing of the sensitivity of the model when the resource demand distributions are changed around their mean values. To achieve this, we modified the values of the coefficients of variation. As discussed in Section 4, four cases were considered: deterministic, hypoexponential, exponential, and hyperexponential. The relative differences between G' and the other four cases are reported in Table 3.

The system model used was not separable due to the presence of the memory. Eliminating this passive resource, we made the model separable, and we investigated the effects of this factor. Table 4 presents the relative differences between the values of the performance indices produced by workload model G' running on the non-separable and separable system models, respectively.

In all the previous runs, the value of the terminal time z was considered exponentially distributed with mean equal to 40 seconds. By modifying the mean terminal time, we obtained different results both for the performance indices and for the accuracy of the workload model itself. We performed two runs: the first with a higher value of z (60 seconds), and the second with a lower value (5 seconds). In the first case, the mean CPU utilization for G decreased from .95947 to .77735; in the latter, it increased to .99838, i.e., the CPU became completely saturated.

The evaluation of the results obtained running the model for G and for G' showed that, when the one system component is saturated, the model is no longer accurate. Indeed, all the relative differences were much higher than those obtained for z=40 s. For example, the difference in the terminal response time was 17.95% (R=46.521, and R'=54.8738).

Using z=60 s, we obtained R= 11.3 and R'=12.1, with a relative difference of about 7%.

6. Conclusions

The results of our simulation runs reported in Section 5 show that, on the whole, the clustering method for workload model design is reasonably accurate in the context of the case examined in our study.

As can be seen in Table 2, the performance of the system model in Fig.2 is not very sensitive to the dispersion of the resource demands of command types around the center of mass of each cluster produced by the *k-means* algorithm (factor A). Replacing workload G with workload model G' causes the terminal throughput rate to change by only about 2.7%, and the terminal response time by only about 3.7%.

On the other hand, the main indices seem to be more sensitive (especially terminal response time) to variations in factor B, the distribution of the resource demands of the commands of each type (see Table 3). The errors would still be, however, within the bounds of acceptability in many performance evaluation studies. Another observation about the results in Table 3 is that an analytic approach to the problems addressed in this research could easily lead to wrong conclusions unless the workload model were based on careful and detailed measurements of a real workload.

The elimination of the passive resource representing the memory in the model in Fig.2 (factor C) produced non-negligible differences in the major performance indices, though these differences were smaller than those due to variations in factor B, and still acceptable in most performance studies. The conclusion one can draw from Tables 2 and 4 is that the accuracy produced by a clustering method is high even when the system is not accurately represented by a separable model.

Finally, in order for clustering to work well, our results indicate that the system should not be completely saturated. High component utilizations are acceptable (our nominal CPU utilization was almost 96%), but they should not be too close to 100%.

On the whole, the sensitivities we found were reasonably low. Thus, we can state that, in at least one practical case,and under the assumptions discussed in this paper, clustering techniques for executable workload model design have been shown to work well.

Acknowledgement

The authors are grateful to Joe Pasquale for his participation in, and invaluable help with, the workload measurement project.

REFERENCES

[Agra76] A.K.Agrawala, J.M.Mohr, and R.M.Bryant, An approach to the workload characterization problem, *Computer* **9**, 6 (June 1976), 18-32.

[Ande73] M.R.Andenberg, *Cluster Analysis for Applications.* Academic Press, New York, 1973.

[Arti78] H.P.Artis, Capacity planning for MVS computer systems, in: D.Ferrari, ed., *Performance of Computer Installations*, North-Holland, Amsterdam, 1978, 25-35.

[Ferr83] D.Ferrari, G.Serazzi, and A.Zeigner, *Measurement and Tuning of Computer Systems.* Prentice-Hall, Englewood Cliffs, 1983.

[Ferr84] D.Ferrari, On the foundations of artificial workload design, Proc. 1984 SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, Cambridge, MA, August 1984, 8-14; also, *Performance Evaluation Review* **12**, 3 (Aug.1984), 8-14.

[Hart75] J.A.Hartigan, *Clustering Algorithms.* Wiley, New York, 1975.

[Lave83] S.S.Lavenberg, Ed., *Computer Performance Modeling Handbook.* Academic Press, New York, 1983.

[Saue82] C.H.Sauer, E.A.McNair, and J.F.Kurose, The Research Queueing Package, Version 2: CMS Users Guide, IBM Res.Rept. RA139 (#41127), April 1982.

Parameters		Clusters							
		1	2	3	4	5	6	7	8
CPU time	mean	.584	.04	30.03	4.47	26.24	2.645	3.85	1.476
	c.v.	.507	.599	1.306	2.388	.499	3.102	1.699	2.588
Memory	mean	27.259	16.13	254.39	158.26	410.22	29.655	330.91	27.035
	c.v.	.205	1.359	.211	1.447	.311	1.561	1.185	.326
i/o blocks	mean	12.81	1.94	150.31	96.78	297.81	15.75	108.79	13.456

Table 1. Centers of mass of the eight clusters.

Performance Indices					
		CPU	I/O 1	I/O 2	Terminals
Utilization	G	.95947	.192	.191	0.
	G'	.96789	.17	.17	0.
Throughput	G	11.09	5.55	5.53	.33516
	G'	9.72	4.86	4.86	.32616
Mean queue length	G	6.23	.242	.24	13.16
	G'	6.49	.206	.204	13.1
Mean queueing time	G	.56	.043	.043	39.28
	G'	.66	.042	.042	40.16
Response time	G	-	-	-	20.416
	G'	-	-	-	21.1626

Table 2. Performance indices produced by workload G and workload model G'.

Relative errors								
	G' vs. DET		G' vs. HPO		G' vs. EXP		G' vs. HPR	
U	.96	%	.59	%	.72	%	.88	%
X	5.94	%	5.1	%	5.48	%	4.36	%
R	18.	%	15.9	%	17.1	%	16.8	%

Table 3. Relative errors in CPU utilization (U), terminal throughput (X), and terminal response time (R) for the different distributions.

	CPU		I/O 1		I/O 2		Terminals	
Utilization	.22	%	12.	%	11.8	%	0.	
Throughput	.51	%	.73	%	.3	%	5.66	%
Mean queue length	8.2	%	1.	%	1.2	%	4.1	%
Mean queueing time	8.1	%	.28	%	.92	%	1.68	%
Response time	-		-		-		13.8	%

Table 4. Relative differences between performance indices produced by a non-separable model and its separable counterpart under workload model G'.

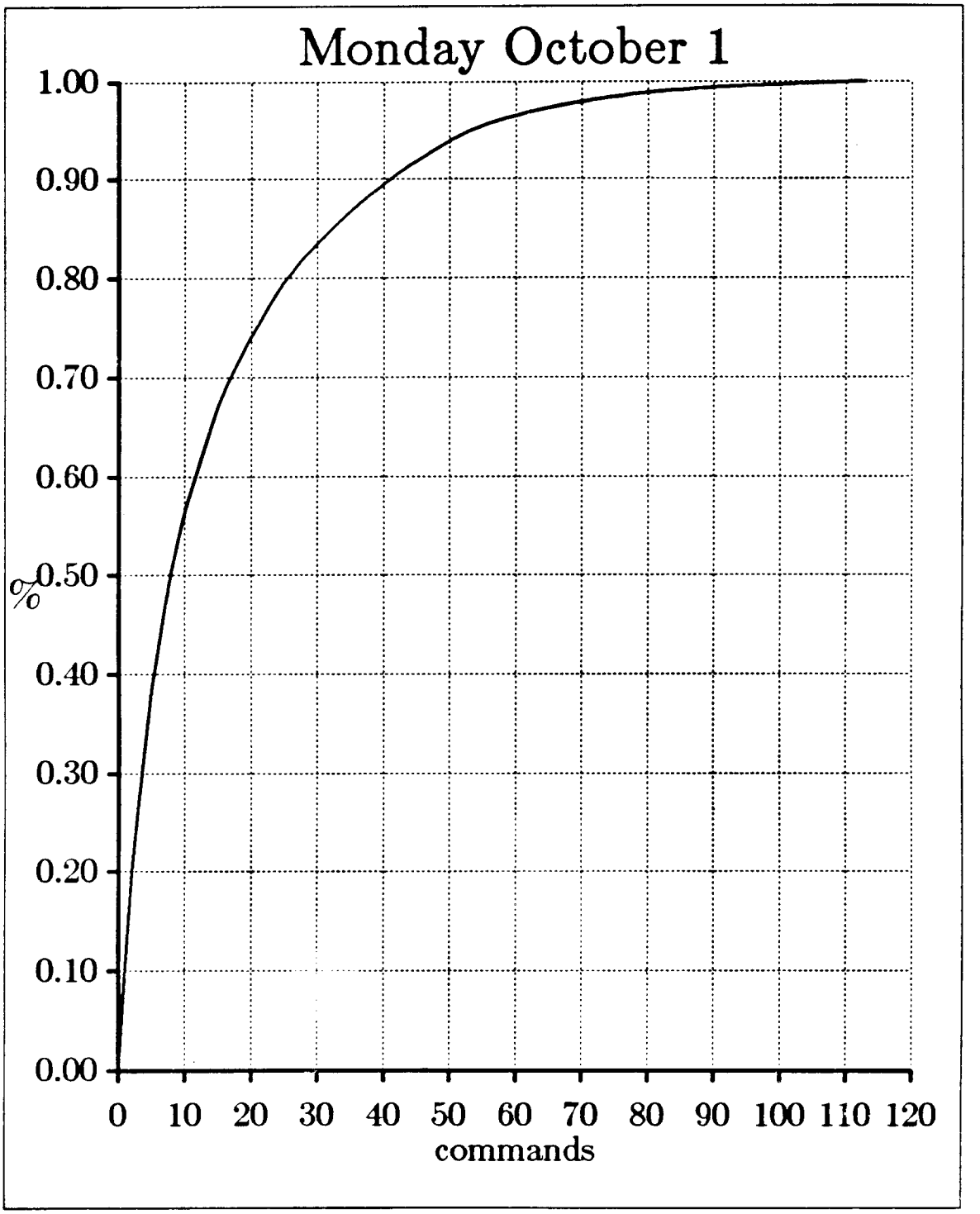

Fig.1. Cumulative frequency distribution of command types.

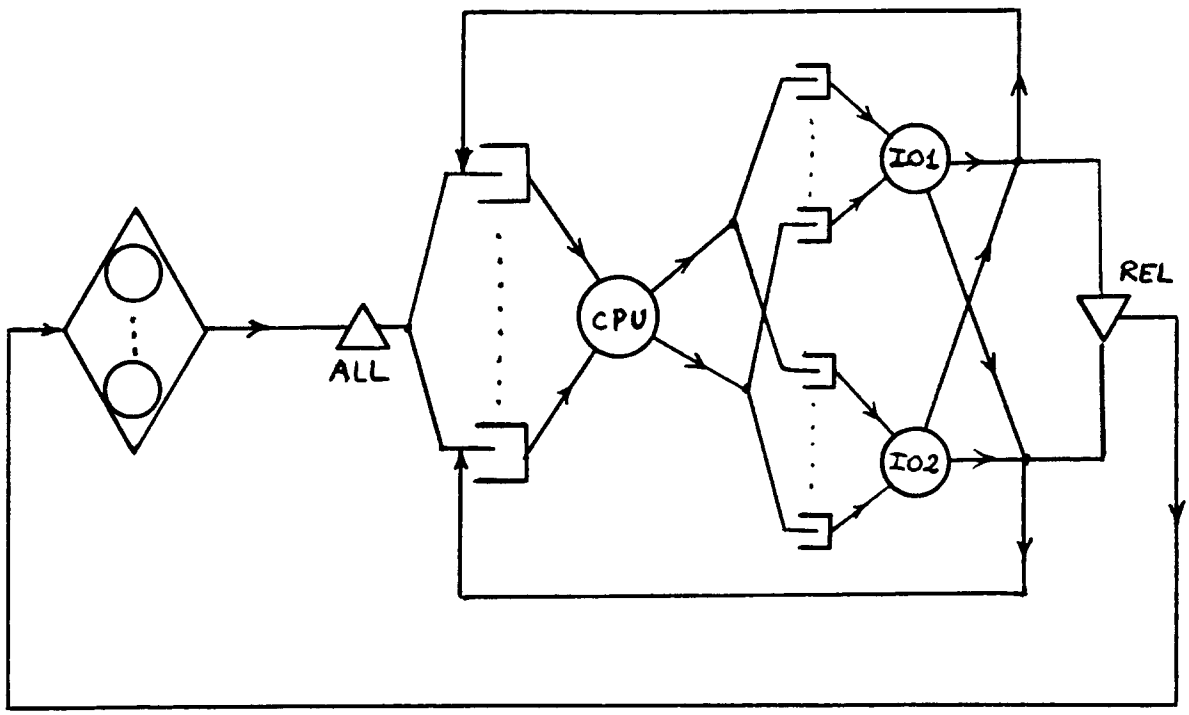

Fig.2. The structure of the system's simulation model.