0 0 0 0 4 7 0 9 2 8 8

UC-32

To be presented at the 1977 SIGMETRICS/CMG
VIII Performance Conference, Washington, DC,
November 29 - December 2, 1977

LBL-6115 Rev. 2 c.1

OBFUSCATORY MEASUREMENT

David F. Stevens

July 25, 1977

Prepared for the U. S. Energy Research and
Development Administration under Contract W-7405-ENG-48

For Reference

Not to be taken from this room

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY

RECEIVED
LAWRENCE
BERKELEY LABORATORY

OCT 17 1977

LIBRARY AND
DOCUMENTS SECTION

LBL-6115 Rev. 2 c.1

LEGAL NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

0 0 0 0 4 7 0 9 2 8 9

LBL-6115
Rev. 2

Obfuscatory Measurement*

David F. Stevens

Lawrence Berkeley Laboratory
University of California
Berkeley, California

July 25, 1977

Abstract

"Obfuscatory measurement" is the practice, deliberate or not, of obscuring the true performance of a system through the use of misleading measures. Many of the traditional and widely-used measures of computer system performance are obfuscatory: they measure the wrong things, the right things wrongly, or nothing at all. Several obfuscatory measures are considered; those aspects of the system they are thought to measure are contrasted with those they actually measure; and alternative measures, which are more meaningful to the user community, are suggested.

Introduction

One of the more common complaints of computer users is that DP management seems to be unaware of real problems. This is partly because user and supplier are ritual antagonists, and one essential characteristic of the good user is a perpetual lack of satisfaction. But it is also

---

* This work was done with support from the United States Energy Research and Development Administration.

partly because DP management sees the operation through the rose-colored glass of obfuscatory measurement, which presents the system in a much more favorable light than that perceived by the user. As a result, DP management cannot understand the users' legitimate distress, and interprets their complaints as mindless nit-picking; the users, on the other hand, see their complaints answered by performance figures which do not reflect their experience, and which lead to no improvement. In order to bridge this communication gap, DP management must adopt a set of measures which illuminate, rather than obfuscate, the true external character of the system.

This paper is an attempt to expose the obfuscatory nature of some of the most widely-used computer performance metrics, and to propose alternative or additional measures which may be used to evaluate the system from the users' point of view.

Availability

One of the strengths of the English language is the ease with which old words can acquire new meanings. This facility can be a mixed blessing during the interregnum when the old and new meanings are both current. "Availability", as currently defined in the Computer Performance Measurement lexicon, is a case in point: simple transference of the old meaning would define "availability" as the fraction of the time the system can be used; whereas the new meaning is synonymous with "uptime": The fraction of scheduled time the system is available to the computer center.

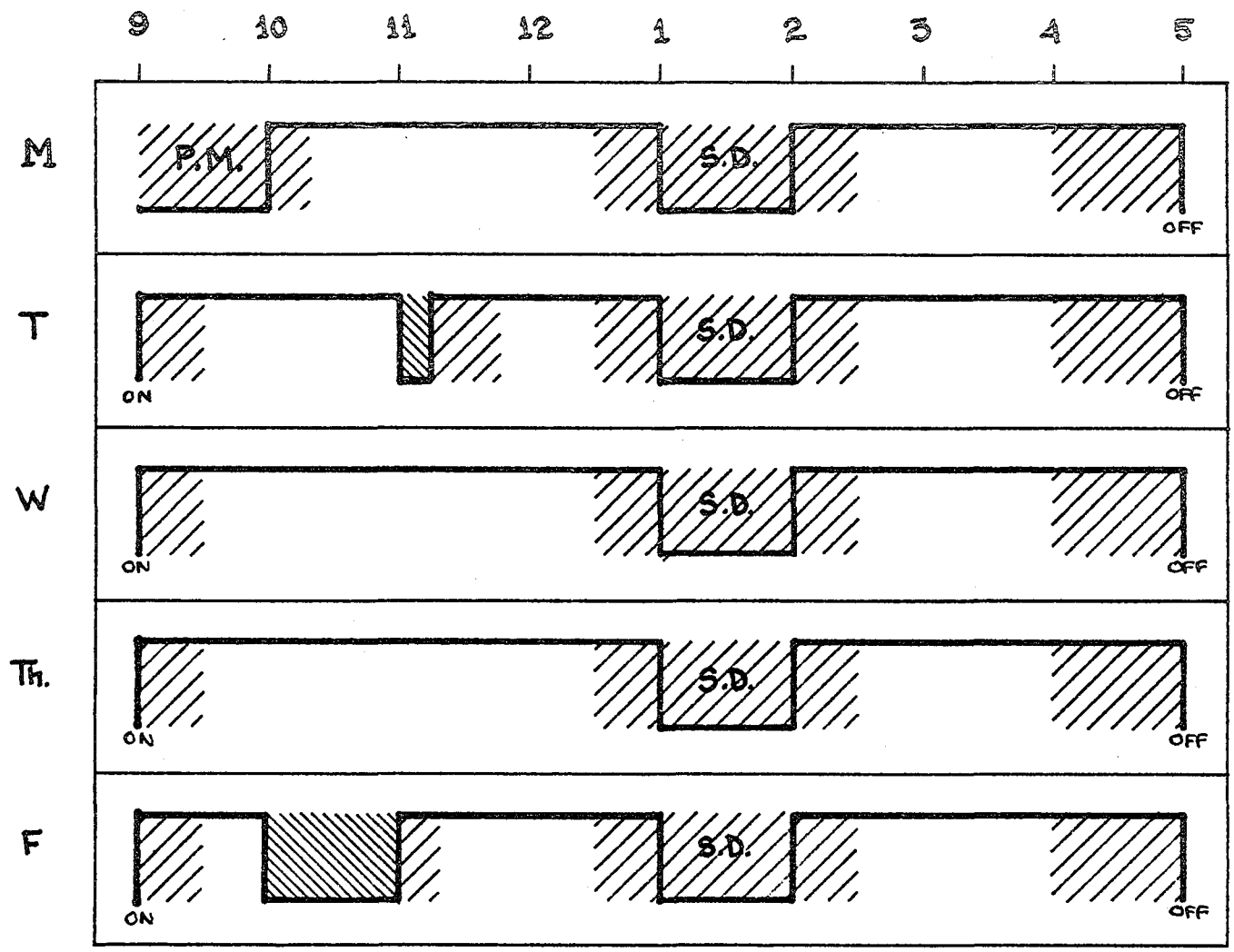
The new meaning differs in three important aspects from the expected meaning:

(1) the time base is reduced, from time (meaning real time) to scheduled time; time devoted to such activities as preventive maintenance or operating system development and installation is excluded from consideration;

(2) the time considered "available" includes many periods of time when the system is not, in fact, fully usable: time devoted to starting the system at the beginning of the day or after an interruption; time devoted to the "run down" before a scheduled interruption;

(3) the time considered "available" also usually includes time devoted to re-running lost or interrupted jobs.

This definition of "availability" is deceptive because these differences combine to give the impression of a much higher level of performance than that seen by the user. As an example, consider the week shown in Exhibit 1. We assume a one-shift operation (9-5, Monday-Friday), with preventive maintenance every Monday 9-10, systems development daily 1-2, and with run-downs (input off) starting daily at 12:30 and 4:00. We will further assume that the system was down for 15 minutes (11:00-11:15) on Tuesday and one hour (10:00-11:00) on Friday, and that all starts and restarts require half an hour. The reported "availability" would be a seemingly respectable 96.3% (32.75 hours out of 34); the user, however, would find the system available (in the sense of able to accept his input) only 48.1% (19.25 hours out of 40)! Discrepancies of this magnitude (which are not uncommon) help to explain the failure of some DP managers to understand their users' cries of anguish and outrage.



	D.P. MGMT. (AVAIL/SCHED)	USER (AVAIL/TOTAL)
M	6/6	3.5/8
T	6.75/7	3.75/8
W	7/7	4.5/8
Th.	7/7	4.5/8
F	6/7	3/8
	32.75/34	19.25/40
	96%	48%

P.M. = PREVENTIVE MAINTENANCE
S.D. = SYSTEM DEVELOPMENT

= DOWN
= INPUT NOT AVAILABLE

EXHIBIT 1.

There are two alternatives to this abuse of the term "availability". One is to keep the measure and change its label to "uptime", but that does not help the DP manager to discover the true availability of his system. The other approach is to keep the label, but to attempt to measure availability from the user's point of view; i.e., that fraction of real time the system will accept input, as indicated in the example.

MTBI (Mean Time Between Interruptions)

"Mean time between interruptions" is a common measure of general system reliability. It can be derived for hardware (by considering only interruptions caused by hardware failure), for software, or for the total system. It is a generalization of a commonly used hardware reliability measure (mean time to failure, or MTF) and has little validity as a computing system reliability measure. It fails to recognize an essential difference between mechanical systems and computing systems: that failures in mechanical systems are caused by wear and fatigue, whereas failures in computing systems are caused by inadequate provision for unexpected input. (It is rare for an overflow on the meter to cause a taxi to crash; it is quite common for an equally trivial overflow to crash a multi-million-dollar computing system.) Increasing length of service is thus a much less significant source of failure than increasing diversity of workload.

The user, in any event, is more interested in the length of the service interval between his job input time and the next suspension of service, scheduled or not, than in MTBI. An accurate determination of this interval is beyond the scope of this paper, but an acceptable substitute is the mean service interval. To see how this compares with MTBI we return to the sample week of Exhibit 1.

As was noted above, the DP manager would claim 32.75 hours of production, and would recognize 2 interruptions (the two down periods). MTBI is calculated by dividing the hours of service by the number of interruptions plus one: in this case, 32.75/3 or 10.9 hours...three times as long as the longest uninterrupted period of service seen by a user! The mean service interval, even giving full credit for the run-down periods, is 2.23 hours (26.75/12).

The mean service interval is a less favorable measure than MTBI (it will always be shorter), but it corresponds much more closely to the user's view of the reliability of the DP facility.

Averages (means) and percentages

It is no coincidence that all of the specific misleading measures discussed here are either means or percentages: means and percentages are the easiest measures to obtain. One must be extremely careful in dealing with means and percentages, however, in order to ensure that they give an accurate and meaningful picture of the state of the system. We have seen in the case of "availability" in particular how the selection of an artificially small base helps produce an artificially high percentage. And while averages are useful, they tend to obscure meaningful detail. Any time one uses averages, one should also, at least occasionally, examine the median and the distribution of the quantities averaged. As an example* of the use of the distribution, let us consider the plot of mean service interval shown in Exhibit 2a. In the absence of additional information, it appears that something bad started happening in August and it cut the

---

* The example is contrived.

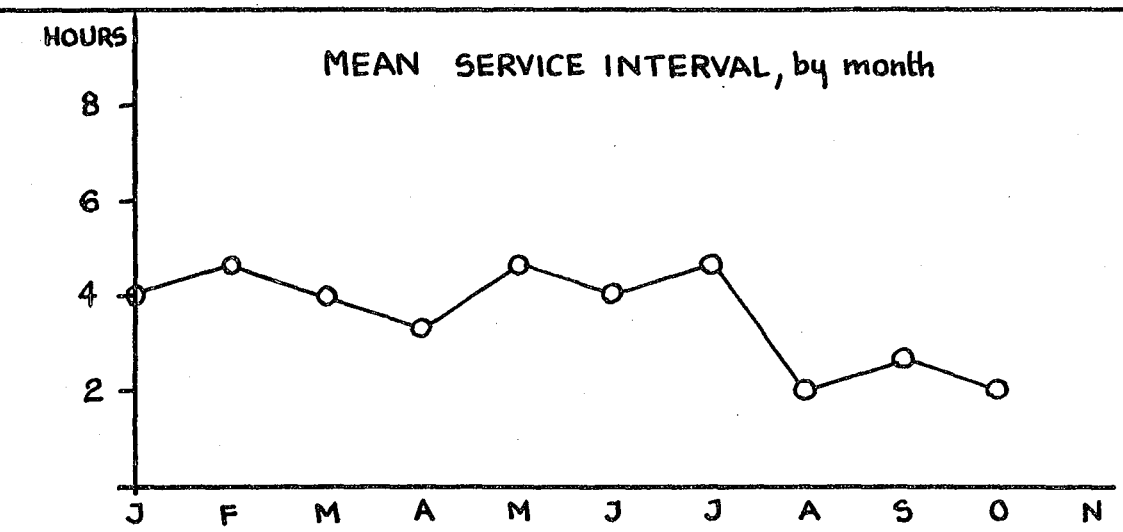


EXHIBIT 2a.

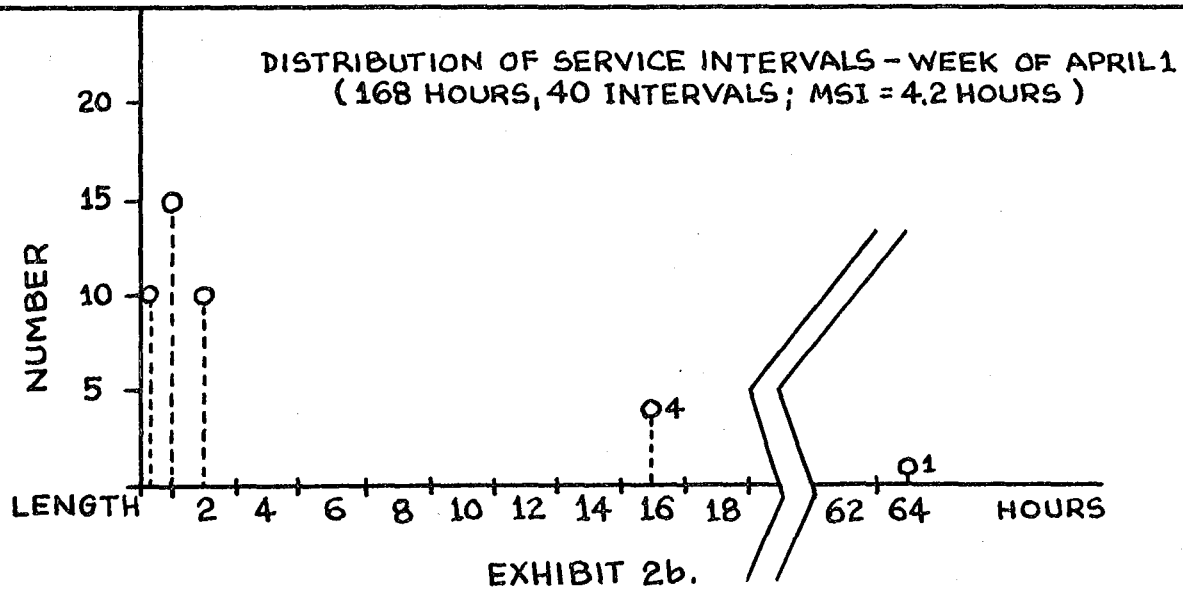


EXHIBIT 2b.

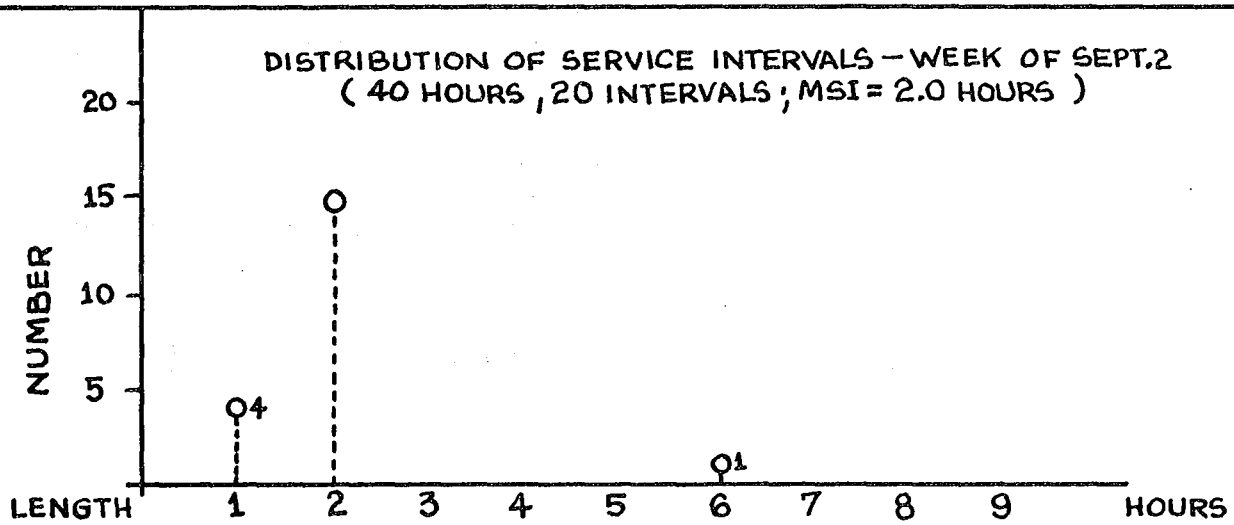


EXHIBIT 2c.

mean service interval in half. In fact, something good started happening in August: the expected service interval during prime shift increased from about 1 hour to about 2 hours. The dip in the measured average was caused by the resulting possibility of eliminating the hitherto necessary, but relatively unused, off-shift rental hours. The true state of affairs is more accurately reported by the distributions in Exhibits 2b and 2c than by the simple averages of 2a.

It should also be clear from this example that one should not wait for an anomaly to appear before examining the distribution of a measured quantity: if the April distribution were not available in September it could not have been compared with the September distribution, and the explanation of the true cause of the drop in the length of the average service interval might have been difficult.

Utilization ("Efficiency")

One problem with raw utilization measures is that they do not tell management by whom, for what, or how effectively the system was used. The situation has improved in recent years with the introduction of separate "system" and "problem" states for CPU utilization, but it remains the case that much of what is called "problem state" is actually system overhead. With respect to channel utilization, the situation is still primitive: channels can be "utilized" even when no data transfer is in progress, and there is no meter to differentiate between system (overhead) channel activity and user channel activity.

Another difficulty with utilization measures is that they reward, if not actually encourage, inefficient usage: if one is being evaluated

on the fullness of her basket she's not likely to spend very much time packing it more effectively.

I do not wish to give the impression that a DP manager should not measure and know the level of utilization of the various devices under her control: quite the contrary, she should have that information at her fingertips at all times. I do wish to discourage the use of utilization measures in the evaluation of the DP function by upper management. "Utilization" is a poor, and often counter-productive, approximation to throughput, which is the proper yardstick.

Unfortunately, there are no very good approximations, unless the workload is very stable. With a constant, periodic workload one can use jobs processed per period; with a strongly production-oriented shop one can use reports produced per period; some installations have defined "computing units"--combinations of CPU time and other resource usage charged to the user--with which they measure throughput. Many measures are possible; the ones selected should reflect the nature and the goals of the installation, and the computing power delivered to the user.

I would like to deplore in passing the use of the word "efficiency" as a synonym for "utilization"--as in the phrase "58% CPU efficiency": not all utilization is efficient. If you must speak of utilization, call it utilization.

Overlap

The desirability of overlap, whether achieved within a single task or via multiprogramming, is widely understood, but effective measures of the depth, or multiplicity, of overlap are not so well know. "Overlap" as usually defined, whether CPU/Channel or Channel/Channel, indicates the

existence of the kind of overlap specified but does not measure its depth. The depth of overlap is the number of simultaneous processes in operation; an acceptable approximation is the sum of the utilizations (expressed as fractions) of all channels plus the CPU. (This is generally not the same as the number of simultaneously active jobs or partitions or tasks in residence.)

This measure shares the major disadvantage of all utilization measures: high system overhead and inefficient use contribute to a "good" score. Nevertheless, it is a good indication of the capabilities of the system, and it should not be allowed to fall too far below the number of active partitions without investigation.

Saturation; capacity

"Percent of saturation" is, like "availability", a misleading abuse of language. "Saturation" is not a measure but a binary condition: a system is or is not saturated; the difference is easy to detect: a saturated system is one which is being given more work to do than it can process under existing restraints, whether or not it's working to capacity, and the only reliable external manifestation is the lengthening input queue.

References to "80% of saturation" really mean 80% of capacity, and are doubly misleading because saturation can be largely independent of capacity. We can best illustrate this by means of an example: Consider a long, narrow footbridge with a capacity of 2000 pounds. Thirty-nine small boys, each weighing less than 50 pounds, would not exhaust its capacity but would surely saturate it (they wouldn't all fit at once) for quite a while. A single man leading an elephant, on the other hand, would not saturate the bridge but would exceed its capacity (see Exhibit 3).

0 0 0 0 4 7 0 9 2 9 4

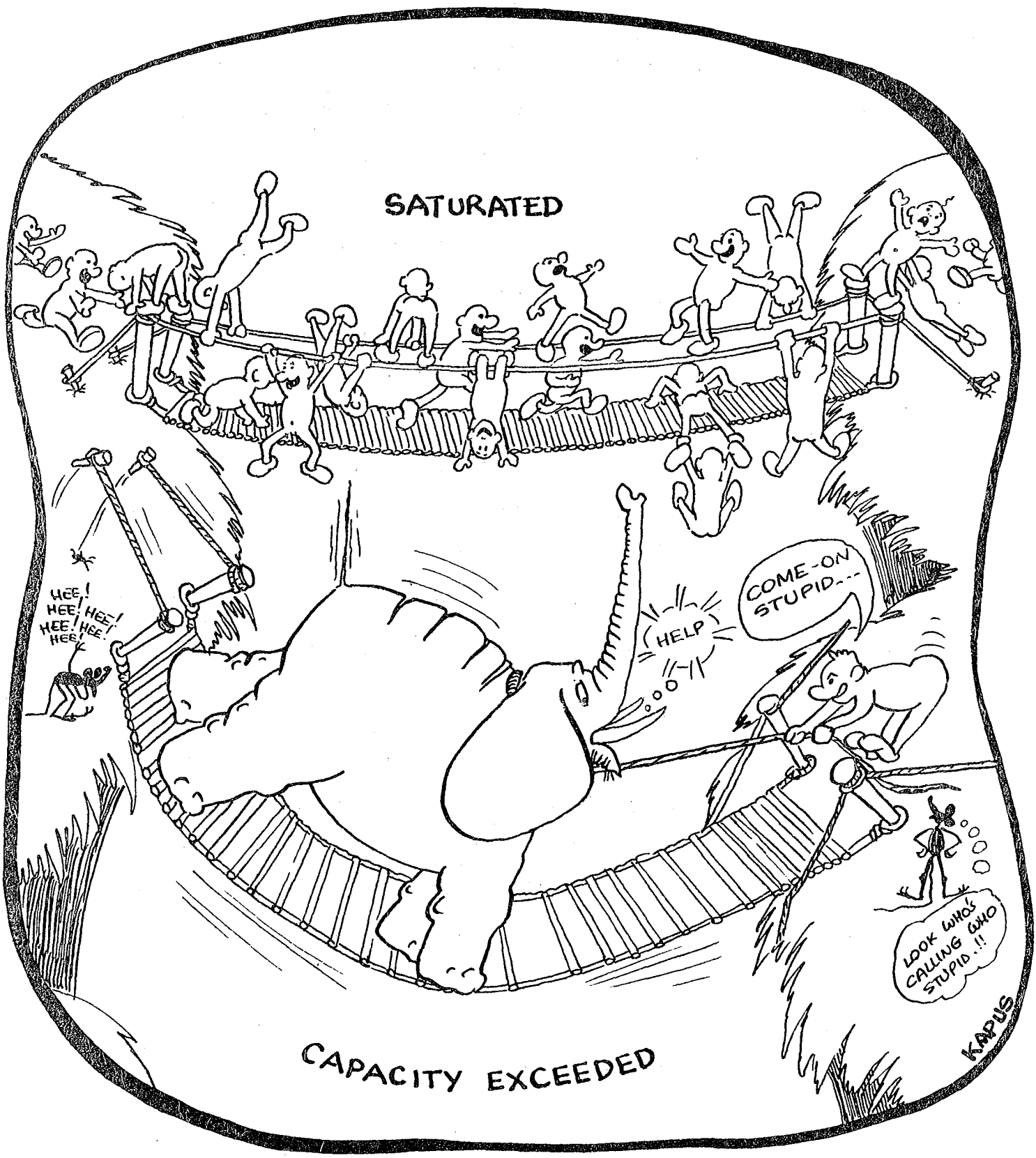


EXHIBIT 3

Attempts to measure throughput as a percentage of capacity are misleading because they ignore the basic fact that "capacity" changes with workload and environment. It is well known that any reasonable multiprogramming system has less capacity when restricted to highly compute-bound jobs than when fed a mixture of compute- and I/O-bound work; it is less well understood that any multiprogramming system strongly dominated by priority considerations has less capacity than a system free to assign requested resources (such as the CPU) in an optimal fashion.[1]

Once more we find ourselves in the utilization-measure trap: a DP manager evaluated on "production achieved as a percentage of capacity" is not strongly motivated to increase the capacity of his system, except by the addition of new equipment. It is clear that the proper measure is "maximum achievable capacity (with a given configuration)".

Interactive response time

"Interactive response time" is another meaningless statistical generalization. For response time measurements to be meaningful it is necessary to know response to what? under what conditions? at what time of day? The situation is further complicated by lack of definitive knowledge about what consitutes optimum response time: folklore exists to support each of the following views:

(a) fastest is best

(b) too fast causes anxiety and increases errors

---

[1] Stevens, On overcoming high-priority paralysis in multiprogramming systems, CACM, August 1968.

(c) too slow causes impatience and increases errors

(d) most constant is best.

It seems clear* that optimal response contains elements of all four, and any measure which encourages one to the exclusion of the others is defective. (The standard "response time" measure encourages (a) to the exclusion of the rest.)

One way to alleviate the deficiencies of the standard measure is to classify the requests and examine the response time by classification. Responses to "trivial" requests (log-off; close a file;...), for instance, should be instantaneous; some requests (log on, say) may take a few seconds, but should instantaneously let the user know what's happening; others (complete searches of large, non-resident files) may justifiably require response times in excess of a minute. Another, partial, approach to the problem is to partition the response time into "process time" and "system wait time"; this, combined with the classification scheme suggested above, will provide the kind of information upon which one can base a rational evaluation of an interactive system.

What does it all mean?

It means that upper management frequently doesn't know the true status of the EDP facility. Sometimes it means that the DP manager himself doesn't know the true status of the DP facility. It is recommended that the measures used to report DP performance be reviewed; any which appear in the left-hand column of Exhibit 3 should be replaced or supplemented

---

* This statement reflects the author's belief that that response is best which most closely matches the expectations of an optimistic user.

with the appropriate measures from the right-hand column. This will assist DP management to see the system as it is seen by the users, and to focus their attention upon those aspects of current performance which the users find most distressing. This, in turn, should help to lessen the mutual distrust which appears to be characteristic of so many user/DP relationships.

0 0 0 0 4 7 0 9 2 9 6

Exhibit 4

OBFUSCATORY MEASURES		RECOMMENDED MEASURES
Common name	What is actually measured	
Availability	Uptime	Availability for general-purpose use, as a percentage of a real time.
MTBI	Mean time to (hardware or software) crash	Mean Service Interval
Utilization	Resource occupancy	Throughput (work delivered)
Efficiency	Resource occupancy	Throughput (work delivered)
Overlap	Existence of overlap	Depth of overlap (the sum of utilization all channels, including CPU)
Saturation	Work as % of capacity	Capacity; existence or not of saturation; throughput (in absolute terms)
Interactive response time	(Nothing)	Process time vs. system wait time, possibly by category

This report was done with support from the United States Energy Research and Development Administration. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the United States Energy Research and Development Administration.

0 0 0 0 4 5 0 2 2 9 4