0 0 0 0 4 6 0 6 0 6 7

Presented at the Interagency Committee
on Automatic Data Processing/Special
Interest Group for Installation
Management, Washington, DC,
December 7, 1976

LBL-5561
c.1

THE COMPUTING UNIT: A MEASURE OF TOTAL COMPUTER UTILIZATION

David F. Stevens and Howard S. White

December 7, 1976

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

LBL-5561
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.

Presented to the Interagency
Committee on Automatic Data
Processing/Special Interest
Group for Installation Management,
Washington, D. C.,
7 December 1976

LBL-5561

THE COMPUTING UNIT: A MEASURE OF TOTAL COMPUTER UTILIZATION*

David F. Stevens and Howard S. White

Computer Center
Lawrence Berkeley Laboratory
University of California
Berkeley, California

7 December 1976

*This work was done with support from the
U.S. Energy Research and Development Administration

0 0 0 0 4 6 0 0 0 6 9

LBL-5661

The Computing Unit: A Measure of Total Computer Utilization*

David F. Stevens and Howard S. White

Computer Center
Lawrence Berkeley Laboratory
University of California
Berkeley, California

7 December 1976

Background

In the early years of computing, the wall clock provided a sufficient tool for measurement of computer usage. Since the computer could run only one job at a time, its total resources were placed at the disposal of that job. It was reasonable to charge the job for the cost of the entire system during the elapsed interval of use, whether or not the job used all of the system resources. Since the operating system could easily keep track of the time when control was transferred to and from the job, the practice of measuring usage by elapsed central processor time was both simple and adequate. (This practice was strongly encouraged by the fact that the cost of the central processor was usually the dominant factor in the cost of the total system.)

With the development of more sophisticated machines capable of being multiprogrammed, we now find many jobs sharing in the simultaneous use of the computer's resources. While only one job has control of a central processor at any instant, a number of jobs are resident in central memory, and may simultaneously be transferring data between memory and various input/output devices. Each of these jobs is executing useful work, and is drawing upon some part of the computer's total resources. Since different jobs use the

---

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

various resources in different proportions, it is unreasonable to continue to base charging upon use of a single resource. Furthermore, the relative cost of the central processor has decreased substantially from earlier years: no longer does the cost of the central processor dominate the cost of the machine. The use of all parts of the computing system should be considered in establishing the cost of a job.

In the broadest of terms, computer resources may be classed as central processor (CPU), memory, and input/output (I/O) devices. Different jobs need greatly differing amounts of each of these resources. Despite the strong folkloric tradition to the contrary, general-purpose scientific environments such as LBL see only a small fraction of the jobs make strong demands for the central processor while using little memory or I/O. Many jobs are I/O limited, while others are characterized by use of large amounts of memory.

One task of a well-designed scheduling algorithm is to select jobs from all those waiting to be performed in such a way that the fullest degree of utilization of the whole computing resource is achieved. The scheduler must meet other constraints regarding priority of processing for various job classes, of course, and therefore only if the computer is saturated by jobs of all priority classes can the scheduler completely meet its goal of fullest machine utilization. Nevertheless, when jobs require only a relatively small fraction of any one computer resource, a good scheduling algorithm can usually fit them together in such a way that all resources are well utilized.

Jobs which demand a major part of one or more resources tend to be mutually incompatible, however, and much more difficult for the scheduler

to fit together into the computer so as to fully utilize all resources. This is particularly true for memory; when one job uses most of the memory it may not be possible to schedule another job concurrently, however nicely it may complement the "wide" job's use of other resources. During this interval, then, the wide job effectively prevents full usage of the computer.

The Computing Unit

The LBL computing unit (CU) was devised to indicate total system resource usage by a job. By measuring total use and charging for all resources, we hope to encourage rational usage of the computer. (Other design criteria, such as simplicity, reproducibility, and historical consistency, contributed to the genesis of the CU, but they are not particularly relevent to this disucssion.) We have also attempted to provide consistent definitions of the CU for the various systems at LBL, so that the same job requires the same number of CU's whether run on the 7600 or one of the 6000-series machines. Since the 7600 is much faster than the 6000's, then the rate at which CU's accumulate must be highly system dependent.

Four kinds of resource usage contribute to the accumulation of CU's on the 7600. Three of these are related to the demands placed on system resources in the areas of CPU, memory, and direct I/O utilization. The fourth measures the station activity associated with indirect I/O and the transmission of jobs to and from the 7600. (The only I/O devices connected directly to the 7600 are disks; all other I/O is staged (spooled) via a 6000 station.) These components are summed to arrive at the total value of work performed for each job:

$$\text{Total CUs} = 3*CP + ITO + \tfrac{1}{2}*BLD + STAGING$$

where CP = CPU time in seconds

ITO = Interference to others, a measure of LCM use

BLD = Number of LCM buffer loads, a measure of direct I/O

STAGING = 6000 machine CUs accumulated by stage jobs.

These components are discussed in more detail in the following.

The CU Dissected

The term measuring CPU use is proportional to the number of seconds accrued during CPU execution of actual job elements. The time accrued during execution of system routines called by job elements, averaging a few percent of the CP, is not presently included in CP.

The term measuring I/O within the 7600 uses the number of LCM buffer loads as its basis for measurement. This number is incremented by the ratio of the block size to the buffer size for each block loaded, so that it is proportional to total words transferred. Jobs can control the scale factor by adjusting the LCM buffer size, but are encouraged to reach optimum value because of the effect of the ITO term relating to total LCM use. (This form of I/O charge encourages users to match their buffer sizes to the problem and reduce disk activity.)

The term measuring staging is required because of the design of the 7600, which has no peripherals of its own. Instead, it relies upon external stations to stage (spool) input and output to its disks. At LBL, these stations are CDC 6600 and 6400 computers which handle the staging activity as part of their regular batch load. They calculate the work involved, in CU's (using the appropriate 6000-series algorithm), and report the total for inclusion with the 7600 job total.

The term measuring LCM usage is founded upon the job's interference to others (ITO in the sense of the "wide job" problem alluded to earlier: jobs requiring extensive LCM decrease the flexibility of the scheduler and reduce the likelihood that it can fit in enough work to occupy the full capacity of the configuration. Because of this, the ITO term is proportional to the job's use of LCM.

But not all wide jobs are bad jobs: some of them utilize the system so fully that the loss of scheduler flexibility is no detriment to total system performance. The ITO term is therefore limited to that portion of system capacity which is forced to remain unused due to the job's memory demands. This unused computing power is determined by comparing the amount of work actually done by the job with the amount of work which a fully-utilized 7600 might be (realistically) expected to produce in the real time required for the job to run alone on the system. We consider that 4 CU's per second is a realistic output for a fully-utilized 7600. (This number was derived from observing that the system, during periods of high productivity, maintains about 83% CPU activity plus 3 buffer loads per second. Plugging these numbers into our CU formula yields 3.99 CU/sec.) The unused computing power is thus

$$4*RT - (3*CP + \tfrac{1}{2}*BLD)$$

where RT is the real time, in seconds.

The real time, in turn, is estimated by using the same two empirical numbers noted above and giving the job the benefit of an assumption of full overlap. Thus

$$RT = \text{Max}\ (CP/.83,\ BLD/3)$$

From this we arrive at

ITO = (unused computing power)*LCM fraction

The calculation of ITO is updated by the operating system each job-second, and whenever the field length is changed. (In self-defense we do not in practice allow ITO to go negative; negative ITO's are replaced by zero.)

Use of the CU

The sum of all the CU's generated by all jobs can be compared with the number of CU's generated by a fully utilized system to give a measure of the degree of saturation of the total system. This is a more meaningful single measure than conventional measures because it is based upon total system usage rather than the utilization of a single resource; we also try to see that it reflects only CU's actually delivered to the user.

A more subtle aspect of the LBL CU is that it is not based solely upon a foundation of resource utilization. To fully understand the significance of this, one must first understand that saturation is largely independent of the notions of resource utilization and excess capacity. Saturation is by definition the state of the system which cannot complete the work submitted in the time available. All other definitions (specifically including CPU -- or any other type of utilization) are meaningless: if you get the work out, you're not saturated; if it stacks up, you are.

Useful as it is, however, the CU is not enough: to obtain full knowledge of your degree of saturation you must also know the balance of the installation: how much each of the components contributes to the total CU count, and which is the bottleneck. Hence you must continue to examine the individual resources as well.

The Weight Problem

We have carefully avoided mentioning one very crucial aspect of the CU-type of charging algorithm: How one arrives at the relative weights to assign to the different components. At LBL, historical continuity has been one of the most compelling considerations: i.e., the relative portions of the cost for CP and I/O usage could not change too drastically from the effect of the previous algorithm. (As a result, we believe that we--in common with most installations--charge too much for CPU time: more than half our CU's are generated by CPU charges.) Other possible considerations are (1) theoretical or empirical considerations (such as using Amdahl's constant...one instruction executed generates one byte of I/O...to generate relative weights for CPU and channels; (2) moral considerations (which weigh heavily the things one should not do); (3) the relative costs of the various components; and (4) component supply and demand considerations. An accurate weighting based upon the last of these criteria would yield the most valid general measure of total system utilization. (But since it would change with changes in job mix and system design, it might vary too rapidly to be popular.)

L'Envoi

We believe the philosophy underlying the LBL Computing Unit is suitable for any computer. It is mildly dependent upon assumptions of maximum machine performance, but the calibration can be verified experimentally. By use of appropriate benchmark runs, a relative scale of computing units can be determined between different computers. But its principal values are that it measures the usage of the system as a whole, not just one piece of it, and that it marks an initial attempt to recognize the distinction between "full utilization" and "saturation".

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