LBID-491
c.1

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Engineering & Technical Services Division

RECEIVED
LAWRENCE
BERKELEY LABORATORY

MAY 24 1982

LIBRARY AND
DOCUMENTS SECTION

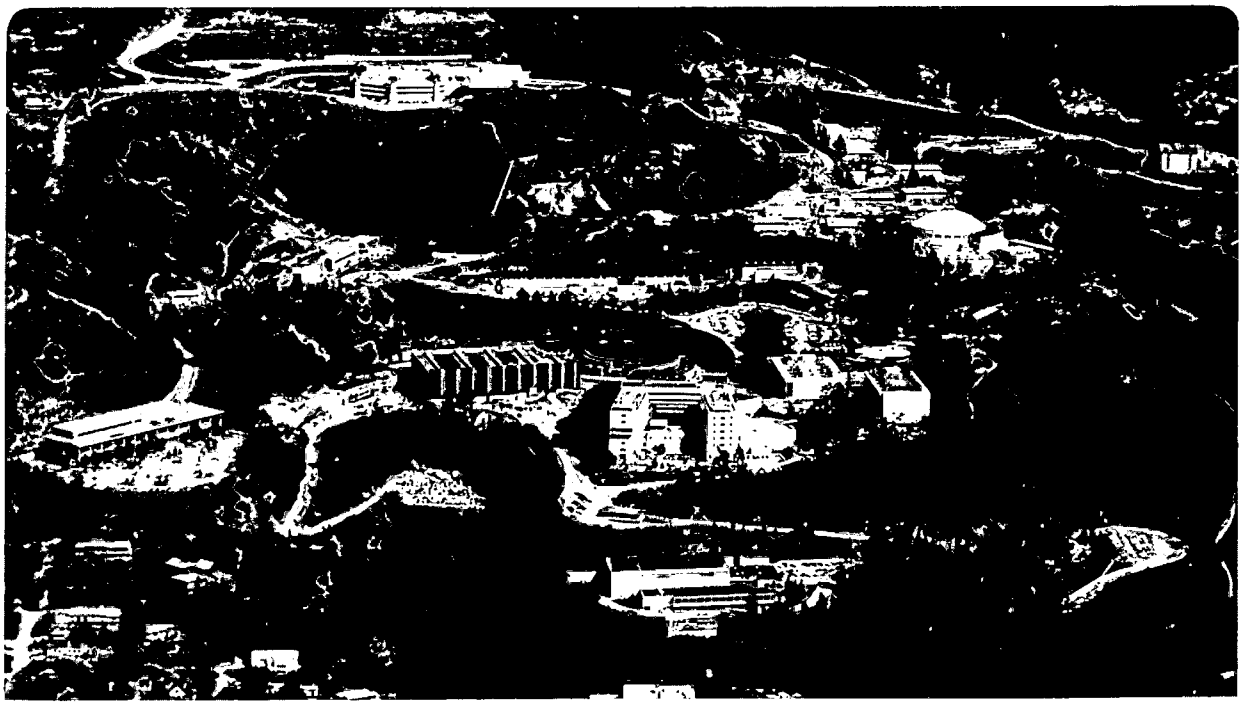
ONE VIEW OF COMPUTING AT THE LAWRENCE BERKELEY LABORATORY

Martin S. Itzkowitz

September 1981

For Reference

Not to be taken from this room



Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

LBID-491
c.1

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal 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. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBID-491

ONE VIEW OF COMPUTING AT THE LAWRENCE BERKELEY LABORATORY*

Martin S. Itzkowitz

Lawrence Berkeley Laboratory
University of California
Berkeley, California

September 1981

* This work was supported by the U.S. Department of Energy under contract No. DE-AC03-76SF00098

September 22, 1981

To: Computer Policy Board

From: Marty Itzkowitz

I would first like to introduce myself. I am in the Software Support and Development group of the Computer Center. I was originally trained as a chemical physicist, with a Ph.D. in chemistry and physics from Cal Tech. I then spent two years as a post-doctoral Fellow in the chemistry department on campus. In 1968, I joined the staff of the laboratory as an applications programmer, working for the Physics, Nuclear Chemistry and Chemical Biodynamics divisions. Following a leave of absence spent at CEA-Saclay, I became a systems programmer in the Computer Center and worked on our local microfilm system. I then was project leader for the 7600 operating system development; following its completion, I spent some time as head of systems programming and consulting. In 1978, I resigned my management position to work on our local HYPERchannel network, and I am currently working on network file transfer services.

For many years I was a user of computing services at various scientific computer centers; more recently, I have been a pusher of those services. From this dual perspective, I would like to present my views on the laboratory's need for computing services and the way in which our computers may best be used.

The Laboratory's Computing Needs

Computers have become essential to scientific research. The tool kit of computing services includes the programs available to the users, the computers and systems on which they run, and the necessary support and operations staff. I believe it essential that we provide as wide a range of powerful, efficient, and easy-to-use software and services as we can afford, so that we can maximize the productivity of the scientific staff.

If appropriate tools are not available, the laboratory's research effort suffers. If a researcher wishes to run a collaborator's program to analyze experimental data, and is compelled to spend several weeks converting that program to the (different) system available locally, time is lost. If a postdoc trying to edit a program is so frustrated by the poor performance of the system that he or she gives up and goes home for the day, the group's research is held up. If a graduate student spends several days tracking down a bug in some program because expert consulting is not available, productivity is lost. More seriously, if the exploration of a particular avenue of research requires the running of a large vector simulation model, and no appropriate system is available to run such a model, the scientist must either go elsewhere or abandon a promising idea.

No single computer system offers a comprehensive enough range of functions to meet all the needs of the laboratory. Just as a research library needs books from various publishers, a research computer center needs machines and programs from a variety of suppliers. The text pro-

cessing services of the UNIX operating system are available on DEC machines; vector FORTRAN codes require vector machines; SPIRES database service runs on an IBM-style machine. Within the Computer Center alone, we currently provide support for our users on six different operating systems; more are contemplated.

All computers come with a set of software (usually for a fee). Other software packages may be purchased or leased from independent suppliers; still others may be developed locally. The price of purchased software reflects the amortization of development costs over a wide set of users; the laboratory pays the full development cost of locally-written software. Wherever a purchased system provides the appropriate set of features, it should be used; likewise, wherever standardization and portability are important, purchased software should be used. On the other hand, wherever the necessary software cannot be purchased, it must be developed locally.

Using standard software is not without cost. License and maintenance fees must be paid. Local expertise must be provided for installing new releases, investigating and reporting system failures and bugs, advising puzzled users, and supplying and deciphering the standard documentation. Moreover, it is important to remember that standard software is just that: it is not prime, nor choice, nor even good. It represents a lowest common denominator, and, as such, may be appalling. IBM's OS/MVS operating system is regarded by some as an archetype of poor human-factors design and implementation, despite the 5000 programmer-years invested in it. Computer manufacturers who supply software have little incentive to produce efficient software; their interests are

better served by selling more and bigger machines.

On the other hand, local system development also is a mixed blessing. Extensive reworking of an operating system represents an investment of hundreds of thousands of dollars and should be considered only where clear benefits of that scale can be seen. Our two locally-rewritten CDC systems provided substantial improvements in both performance and features relative to the then-available standard versions, repaying development costs many times over. We are now paying an additional price: we cannot install the new FORTRAN-77 compiler under our obsolescent system. (On the other hand, standard software cannot support most of our obsolescent I/O devices.)

There are, however, some areas where local software development is clearly appropriate and cost-effective. Modular extensions to a standard system can be made with relatively little risk and significant benefit. GSS, a remarkably cheap and effective archival storage system, and our local libraries for device- and machine-independent graphics are two examples.

The preceeding discussion has centered on computing services provided in support of scientific research. Clerical and administrative support of the laboratory require other computing services, as does the laboratory's research in computer science. Each of these has its own specialized needs.

In summary, I believe the laboratory needs a wide variety of computers and software packages, purchased where possible, and developed locally where not.

Using the Computers

I believe that the laboratory can best use its computers by placing them in a centralized facility, and interconnecting them with a local network. A significant objection to centralization centers on the issue of control and allocation of the centralized facility. The sense in which I use centralization refers only to the support and operation of the computers. Control and allocation are matters of policy which need not and should not be set within the central facility.

A principal benefit of centralization derives from economies of scale. Large machines provide more power per unit cost than small machines. The Computer Center must sooner or later replace the 6000's with more modern machines to provide staging support; the wide range of peripherals available for IBM-style machines makes one of them an obvious possibility. Similarly, the wide range of commercial software available makes an IBM-style machine a reasonable candidate for administrative data processing. Quite possibly, a combined procurement of one large machine could provide both staging services and ADP services more cheaply than separate procurements.

Centralization of operations allows machines to be run more cheaply. The same operations staff can handle more machines if all of their tape drives and printers are in one place, than if each machine is in a different building.

Certainly, some computing services should not be centralized: a real-time data collection system needs to be near the instrument it is monitoring. Central support services for real-time systems are

necessary, however. Convenient means need be provided to transfer data between such systems and the central facility, either by various magnetic media, or by extension of network services through central gateway machines out to the remote systems.

Interconnection of computers provides many benefits. Each machine can provide those services that it performs best, while any user, no matter where he or she is running, can obtain the full range of services. Each system is effectively extended with at least some of the features of the others; the whole becomes greater than the sum of its parts.

On the surface, such an approach appears expensive. Each different machine and operating system requires support and maintenance, even when no local modifications to the operating system or standard programs are made. In addition, heterogeneous local networks are at the leading edge of computing technology; hardware is expensive and no commercial software is available. Fortunately, the computer center has already made a substantial portion of the investment necessary for such a system. The hardware and low-level software are running now; within the next year we expect to provide GSS and print service for all machines in the local network.

Networking allows improved performance on each of the separate computer systems. A computer system is not a simple resource; it is a collection of quasi-independent subsystems. These include a central processor, memory, I/O channels, peripheral devices, mass storage, etc. Optimal usage of a machine occurs when all of its subsystems are in use at all times. Consequently, the matching of the set of resources

available to the set of tasks to be performed may profoundly alter the performance, and thereby the effective power of the system. The "impedance" of the load must match the characteristics of the system.

For example, the CDC 7600 is a machine that is designed to perform numerical calculations efficiently; the hardware architecture and the design of the operating system are tailored for number crunching. A 7600 can be programmed to do interactive text editing. It can then support as many interactive users as several PDP 11/70's whose combined cost is an order of magnitude smaller, but it can only crunch half as many numbers. You have sold half your 7600 for a mess of PDP-11/70's.

In general, each machine and operating system reflects a series of design choices which make some tasks efficient and others wasteful. A machine which performs only those tasks for which it is well suited is, in a very real sense, a larger and more powerful machine than one which spends some of its time on chores for which it is inefficient. Furthermore, a single machine performs at least some of its operations serially; a network of independent machines runs in parallel.

Another benefit of a network is that costly hardware may be shared. A high-quality printer or tape drive is very expensive: so much so that one is difficult to justify for a single VAX. If it can be shared among several VAX's and other machines, it provides both better and cheaper service. Sophisticated devices such as a color microfilm recorder or mass storage device simply cannot be justified for a single system; some cannot even be run on such systems.

To summarize, I believe that the computing needs of the laboratory are best met by a central computer facility consisting of a

heterogeneous network of computers, and that such a facility is both the most powerful and the most cost-effective way to support the scientific mission of the laboratory.

Thank you for your consideration.

MS Hafitz

This report was done with support from the Department of Energy. 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 Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720