LA-UR -82-2651

Conf-821202--4

LA-UR--82-2651

DE83 000648

TITLE BENCHMARKING YOUR BENCHMARKS: A USER'S PERSPECTIVE

AUTHOR(S) Richard Brice

MASTER

SUBMITTED TO CMG XIII
San Diego, CA
December 13-17, 1982

DISCLAIMER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Los Alamos
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

BENCHMARKING YOUR BENCHMARKS: A USER'S PERSPECTIVE

Richard Brice
Los Alamos National Laboratory

ABSTRACT

This paper is intended for anyone faced with the responsibility for computer hardware or software selection. The content is biased toward administrative considerations, although some technical issues are presented. The basic goal is to promote a role for benchmarking in the computer acquisition process that is significantly broader than that commonly employed. We attempt to do this by showing how a thorough benchmarking effort (in the usual sense) did not provide sufficient information to accurately predict user satisfaction and productivity. We describe other measures of a system's properties that should be included in benchmarking.

1. INTRODUCTION

Computer system benchmarking has gained wide acceptance as a useful approach for predicting the capacity of a new computer system. We view benchmarking as one of a valuable set of tools and methods whereby users, system staff, applications programmers, and administrators can gain some measure of confidence that a computer system will perform the desired functions. The literature contains numerous well-documented examples of benchmark studies [1-5].

A study aimed at predicting the performance of a computer system is thought to be well done if both the workload(s) and the system response(s) are correctly modeled. Such studies are extremely valuable in the system selection process, however they are usually incomplete because they concentrate on demands for computer resources and ignore the human element involved in computing. This omission is understandable, because good models of the "human element" are in short supply and are difficult to define [6-12].

In this paper we list a set of questions that should be answered through some form of benchmarking process but which, in practice, are often addressed informally if at all. We describe some of the problems and frustrations in a computer resource approach to evaluation by following a benchmarking activity through several years and four significantly different levels of benchmark sophistication. The benchmarking was done by the Computing Division of the Los Alamos National Laboratory to evaluate the suitability of the Cray Research, Inc. (CRI) Cray-1 computer and its software to support the Laboratory's computing needs.

2. QUESTIONS FOR BENCHMARKING STUDIES

Most benchmark studies could be significantly improved if they also addressed what the author feels are the real reasons for acquiring new systems:

(1) improved user productivity and
(2) increased user satisfaction.

The assumption is often made that a faster CPU, larger main memory, and more powerful peripherals are what is needed; in other words, improving system performance will improve user productivity and increase user satisfaction. In fact, however, benchmark studies usually provide almost no information about the contribution of a new system toward meeting these two goals. Instead, because of the excellent job they do in assessing hardware capabilities, these studies often lead to overemphasis on the hardware aspects of an acquisition. Examples of other questions that need to be addressed are:

1. How reliable will the system software be?
2. How easy will it be to isolate system or user software failures and fix them?
3. How easy will it be to transport applications into and out of the system and upgrade to the next system we acquire?
4. How easy will it be to merge this new capability with existing capabilities?
5. How rich is the system utility software? How easy is it to use?
6. How easy is it to use system calls, JCL, file management?
7. How representative of the workload are the benchmarks?

Answers to these seven questions seem fundamentally important to any speculation of how productive and satisfied the users may be. Question 7 can be quantified directly in those environments that have a well-defined and slowly evolving workload. There is an ongoing effort at Los Alamos to ensure that our benchmarks do represent the workload. Answers to Questions 1-6 are inherently less specific; however, some approaches to this form of benchmarking have been described.

Pearson describes a questionnaire used to collect user satisfaction data [13]. The data can be used to parameterize the formula

$$S_i = \sum_j W_{ij} * R_{ij} \qquad (1)$$

where

S_i = the satisfaction of user i,
W_{ij} = the weight User i gives to performance factor j, and
R_{ij} = the raw score User i gives to factor j.

Deese describes application of the questionnaire with some positive results [14]. This approach is a step in the right direction, but suffers three drawbacks. First, when the S_i are summed (or averaged), some weight is needed to represent value of the user to the survey, for example, technical competence or programmatic importance. Second, the measure is not designed to aid in evaluation of proposed, but not yet implemented systems. Finally, the application of weights to nonquantifiable things, such as Questions 3-6, is likely to produce questionable results.

In a recent attempt to determine weight values for factors in a proposed system, few of the factors gained consensus [15]. The scores for most of the factors varied so widely that no prioritization could be found that did justice to all respondents. Fortunately, iteration and compromise eventually allowed an acceptable prioritization to be produced.

Another approach recommended by Brooks [16] and Brittan [17] is to use a series of successively more complete prototypes that converge to the desired system. They claim a near certainty of failure to satisfy the user with the first system delivered. Furthermore, they point out that in practice a series of prototypes will occur anyhow and that the user should be a party to this process from the outset. Two examples of this approach, one modest and the other quite large in scope, are described in Refs. 17 and 18.

In the following discussion of the Cray-1 evaluation at the Los Alamos National Laboratory, we describe our experiences in evaluating the Cray-1 and some of the problems that resulted from insufficient understanding of the human element.

3. CRAY-1 EVALUATION

3.1 BACKGROUND

The Los Alamos National Laboratory is a research and development laboratory operated by the University of California for the United States Department of Energy. The Laboratory role involves weapons and energy-related projects. Many of these projects require enormous computing power. This power is provided by a network of computers at the Laboratory that includes four Cray Research, Inc. (CRI) Cray-1 computers, four Control Data Corporation (CDC) 7600s, and several CDC Cyber and 6000-series machines.

Although the Laboratory workload has never been thoroughly characterized, it is generally perceived to consist mostly of large programs that dominate production (night and weekend) periods and user code development that dominates the prime (weekday) shift. By large programs we mean codes that are typified by some or all of the following requirements:

- several thousand seconds of CPU time on a large scientific computer,
- multiple megabytes of main memory, and
- billions of words of high-speed (disk) I/O per run.

3.2 LEVELS OF BENCHMARKS

We have used the following four levels of benchmarks at the Los Alamos National Laboratory to evaluate the performance of the Cray-1s and their associated software.

Level 1 - Small, hand-coded and optimized machine language programs (called kernels) selected to represent important portions of the projected workload.
Level 2 - Programs or program segments written in Fortran (more elaborate kernels).
Level 3 - Programs selected from current workload.
Level 4 - The actual user workload, which evolves over time.

3.3 CRAY-1 BENCHMARKING - LEVEL 1

A set of level 1 benchmarks that would quantify the improvement in CPU power to be expected from the CRAY-1 was implemented and run by the Los Alamos Computing Division during early 1976. For one set of results see Keller [20,21].

Keller's work was very carefully done and thoroughly documented. We extract here just his observation that a carefully selected set of kernels would run from two to five times faster on the Cray-1 than on the CDC 7600. We further note that Keller's predictions were later substantiated by benchmarks at Levels 2 and 3.

One inevitable effect of these benchmark results was elevation of user expectations of work they could accomplish using the faster machine. Needless to say, the benchmarks in Refs. 20 and 21 were not actual codes, they were not compiled by an actual compiler (with its inherent inefficiencies), and they were not run under a production operating system (with its inherent overhead).

3.4 CRAY-1 BENCHMARKING - LEVEL 2

Because compilers, utility programs, and other system software for a new class of supercomputers often lack capability and reliability, the decision was made to construct a local version of a Fortran compiler (called XFC) that would run on the CDC 7600 and generate code for the Cray-1. Also, an operating system, DEMOS [22], was developed locally, partly because of the vendor's modest plans for such software and partly to retain local control over the concerns expressed by Questions 1-6 stated earlier.

Preliminary versions of the vendor compiler (CFT), the cross compiler (XFC), and the vendor operating system (BOS) became available in 1978. Table 1 presents some Level 2 benchmark results using this software. Columns 2 and 3 represent new capability on the Cray-1 using the BOS operating system and either the XFC or the CFT compiler. Columns 4-7 represent existing capability on the Los Alamos CDC 7600s using the Livermore Time Sharing System (LTSS), the SLOPE2 subsystem, and the FTN1 and FTN2 compilers. Details of the benchmark codes are found in Ref. 19.

The benchmarks used to produce the data of Table 1 are primarily intended to measure hardware performance on certain well specified operations. They were not developed to measure differences in software, for example, compilers and operating systems. However, it is tempting to draw conclusions about relative performance of system software from these measures.

Table 1 shows that the Cray-1 running BOS/CFT was significantly slower than the existing 7600s in 3 of the benchmarks, only slightly faster in 4 of the benchmarks, and much faster in 3 of the 10 benchmarks. The BOS/XFC combination performed much better, being faster in 7 of 10 benchmarks and slower in only 1 of them.

Performance ratios of new capability to existing capability can be determined for individual benchmarks or for all collectively. Choosing the best value from columns 4-7 for each benchmark and summing the run times gives an overall performance improvement ratio of 1.93 for BOS/XFC over the best CDC 7600 capability. The individual performance ratios range from a high of 6.1 to a low of 0.59. The performance ratios were sufficiently large to indicate that the Cray-1 might provide significant improvement in computing power.

Table 1. Run Times in Seconds for 10 Benchmark Programs, July 1978

Bench-mark	BOS XFC	BOS CFT	LTSS FTN1	LTSS FTN2	SLOPE2 FTN1	SLOPE2 FTN2
1	28.60	64.83	49.32	28.75	49.16	29.15
2	15.91	49.62	34.74	33.68	39.54	37.83
3	46.67	31.80	56.51	52.39	55.80	49.09
4	26.90	30.27	50.81	50.00	53.02	53.10
5	150.77	269.08	104.71	90.06	98.45	88.36
6	35.94	30.91	62.50	41.92	6[illegible].63	42.74
7	79.60	210.20	402.40	363.20	403.40	366.40
8	4.53	3.24	24.30	21.00	22.90	20.90
9	14.02	20.66	102.60	86.40	97.20	85.79
10	5.02	3.61	31.91	23.25	32.87	24.14
Total	407.69	714.22	919.80	745.65	916.17	707.50

Because the results in Table 1 do not suggest which is the clear-cut winning combination, it would be useful to have a function such as

$$f = \sum_i Ri*Pi*Wi \qquad (2)$$

where

- R_i is the performance ratio of benchmark i from Table 1,
- W_i is the workload fraction represented by the benchmark,
- P_i is the probability that Wi will be valid for an appreciable time, and
- i ranges over the benchmarks.

Using function f, we could arrive at a better determination of the improvement to be expected in the actual workload. The existence of such functions is implicit in the use of benchmarks, but frequently the function variables do not have well-established values. The Los Alamos Computing Division is currently implementing a project that will automate (to the extent possible) collection of values for the Wi in Eq. (2) [23].

Note that we have summed the times of the individual benchmarks in both Table 1 and Table 2. This essentially represents using Eq. (2) with a value of 1.0 Pi and a value of 0.1 for the Wi.

Definition of a function that quantifies answers to Questions 1-6 is more difficult. The variables of the function would probably be difficult, if not impossible, to quantify; for example, how much effort is acceptable to a user when trying to use the system?

Three new operating systems for the Cray-1 became available in late 1978. The Cray Time Sharing System (CTSS), developed at the Lawrence Livermore National Laboratory (LLNL), is an upgrade of LTSS (also developed at LLNL). COS is the vendor system and successor to BOS. DEMOS was developed at Los Alamos.

There were enough supporters of each operating system and compiler to justify further study, so another round of Level 2 benchmarks were produced. Some of the results are shown in Table 2.

Table 2. Run Times on Potential Systems and Compilers, July 1978-May 1979

Benchmark	BOS XFC	BOS CFT	DEMOS XFC	DEMOS NEWCFT	COS CFT	CTSS CFT	CTSS NEWCFT
1	28.60	64.83	30.48	58.80	64.85	66.86	59.34
2	15.91	49.62	17.68	43.76	50.65	--	23.50
3	46.67	31.80	50.67	22.56	31.99	32.59	22.84
4	26.90	30.27	26.21	27.12	30.49	31.08	29.58
5	150.77	269.08	153.29	215.05	268.03	274.52	225.19
6	35.94	30.91	34.12	19.13	29.25	30.27	20.90
7	79.60	210.20	92.73	142.30	209.80	211.13	140.42
8	4.53	3.24	4.53	2.92	3.25	3.46	3.05
9	14.02	20.66	24.32	16.69	20.64	20.68	16.91
10	5.02	3.61	3.94	.41	2.61	3.53	3.41
Total	407.90	714.20	437.90	557.30	711.50	697.60	544.10

The results in Table 2 are mixed. The smallest over-all run time belongs to BOS/XFC, however considerations unrelated to performance render this combination unacceptable. For example, XFC does not run on the Cray-1, and BOS is a rudimentary system that lacks numerous, necessary features.

If BOS is eliminated, the two DEMOS combinations each capture best performance in five cases with DEMOS/XFC having the best overall performance.

If XFC is eliminated because of the difficulty of compiling on one machine and running on another, DEMOS/NEWCFT is best in 7 of 10 cases but CTSS/NEWCFT has the best overall time. This indicated that DEMOS/XFC and CFT still had the best performance, but that CTSS was rapidly evolving. The CFT compiler was in development as these tests were carried out. Some performance differences are due to different versions of the compiler. NEWCFT is used to distinguish a significantly different release of the compiler from previous releases.

The benchmarks that produced the results shown in Table 2 were primarily CPU intensive. A thorough set of I/O benchmark studies also were carried out by the Los Alamos Computing Division, and the results are reported in Ref. 24.

Tables 3 and 4 contain a sample of the I/O measurement data discussed in Ref. 24. Numerous other tests were performed for both DEMOS and CTSS to provide a basis for systems comparison and also to support users who were trying to tune their codes to the two systems.

Table 3. Demos Multichannel Read/Write Tests, August 19[illegible]

Read Transfer Rates per Channel (kword/s)

Buffer Size x 512 words	Number of Channels 1	2	3	4
3	71	71	65	51
9	172	157	156	173
18	249	259	245	238
90	381	368	389	367

Write Transfer Rates Per Channel (kword/s)

Buffer Size x 512 words	Number of Channels 1	2	3	4
3	140	103	69	87
9	173	167	166	168
18	252	251	244	251
90	386	374	381	373

Table 4. CTSS Multichannel Read/Write Tests

Read Transfer Rates per Channel (kword/s)

Buffer Size x 512 words	Number of Channels 1	2	3	4
3	79	79	78	78
9	178	179	178	179
18	263	262	263	262
90	425	422	474	422

Write Transfer Rates Per Channel (kword/s)

Buffer Size x 512 words	Number of Channels 1	2	3	4
3	497	413	333	199
9	497	446	484	440
18	495	495	486	494
90	497	496	494	495

3.5 CRAY-1 BENCHMARKING - LEVEL 3

The effort required to convert one large user code to any of the possible combinations of system software is typically measured in man years. To convert numerous user codes to all or even several combinations is just not realistic. Thus we continued with the choice of DEMOS/CFT with very little benefit from Level 3 benchmarks.

As users converted their codes to the chosen system, a great deal of Level 3 results poured in. Most of it was ad hoc and poorly documented and much of it was bad news. Generally, the initial complaints were that the chosen system would not perform as well as one of the others and that programs were not running as well as the Level 2 results predicted. Some of the difficulty was predicted by Table 2, which suggests that none of the candidate systems could match BOS/XFC in performance. The performance problems were usually fixed once identified; see Ref. 25 for detailed examples of the effort invested in I/O performance.

As the system stabilized, the users' programs began to perform at a more acceptable level, and the complaints shifted from performance problems toward the quality of the human interface. Samples selected from a list of complaints compiled from user memos are listed below.

1. Not able to interact with production jobs.
2. Job time limits not deterministic.
3. Job termination procedures inadequate.
4. Too cumbersome to run short checkout jobs.
5. Different environments for users and system developers.
6. A cumbersome, inefficient user interface.
7. Much file conversion between front-end and back-end machines.
8. Operating system is too large and correction of problems often increases its size.
9. Files are not automatically purged after some time limit.
10. The Fortran library is deficient.
11. The utilities are inadequate.
12. Some jobs are not recovered after a system crash.

Unfortunately, we had no benchmarks that would quantify how well the other systems would perform with respect to these and the other complaints.

3.6 CRAY-1 BENCHMARKING - LEVEL 4

Despite attempts to improve DEMOS by addressing complaints like those above, user dissatisfaction continued. At the end of one year of operation, the decision was made to replace DEMOS with CTSS. The decision was based partly on performance issues but to a large extent was influenced by issues suggested by Questions 3-6 as stated earlier. Namely, the users had gained a lot of experience in the use of LTSS (the predecessor to CTSS) and they found it difficult to adapt to DEMOS, which presented a significantly different interface.

During DEMOS's year as the Cray-1 operating system, numerous measurements were made to determine the nature of the user workload and how well the system was responding. These measurements constitute our Level 4 benchmarks. They consist of the observable effects of real users doing real work and do not represent repeatable experiments. We relied, therefore, on averages to evaluate how well the system was responding to user demands for service and also to determine the nature of the demands. Table 5 contains data averaged over several days for some of the performance measures we collected.

CTSS became fully operational at Los Alamos in early 1980. The conversion of user codes to CTSS is virtually complete; however, only recently has the user workload become sufficient to permit meaningful measurement of the system activity. Some of the CTSS measurements are shown in Table 6. The values in Tables 5 and 6 are not directly comparable for two reasons:

1. Some operating system functions charged to the DEMOS system are charged to the user on CTSS. The actual fraction of the CPU cycles that these charges represent is probably about half of those charged to the DEMOS system.
2. The degree to which the workloads on the two systems differ is not known. This reason is more serious because it involves the nature of the two workloads.

In spite of a few problems, many users have expressed satisfaction with the CTSS system. Expressions of satisfaction often do not include much detail, and we do not have a long list of features liked by users. Most of the complaints listed before have not recurred; so we assume that those issues are now at least acceptable to CTSS users.

Table 5. Typical DEMOS Performance Data

Shift	User CPU Cycles	I/O Idle* CPU Cycles	System CPU Cycles	User I/O Byte/s
Production	85.6%	9.2%	5.1%	200,000
Prime	58.6%	21.6%	19.4%	440,000

*The term I/O idle in Tables 4 and 5 represents a measure of the CPU cycles lost because all memory resident processes were waiting for I/O.

Table 6. Typical CTSS Performance Data

Shift	User CPU Cycles	I/O Idle CPU Cycles	System CPU Cycles	User I/O Byte/s
Production	76.5%	22.0%	2.4%	960,000
Prime	79.5%	15.0%	6.5%	528,000

4. CONCLUSIONS

It was difficult to extract meaningful comparisons of our benchmarks at different levels due to the reasons mentioned in the previous sections. Those at Levels 1 and 2 seem to be consistent. The Level 3 results, although less well documented, seem also to be in agreement. We are unsure how to extract Level 4 measurements that can be compared with the lower level results without better values for the workload mix, e.g., the variables of Eq. (2).

The users were dissatisfied and complained that it was difficult to be productive using DEMOS; that is, the assumptions of productivity and satisfaction did not follow from the good benchmark results at Levels 1-3. Performance oriented benchmarks, although important, proved in this case to be poor predictors of user satisfaction and productivity. We suspect that addition of benchmarks that address Questions 1-6 directly would provide a better base for prediction even if their only contribution was to alert users to the rough road ahead.

The answers to Questions like 1-6 are gradually accumulating in a variety of places, such as trouble reports, requests for system changes, and in the vast body of user experience. It is unlikely that the answers will be well documented. It may also be that different organizations will arrive at different answers. We believe this to be typical of large software projects.

In spite of the frustrations presented here, the Cray-1 has played a key role in support of our operation, and we continue to have a high regard for our benchmarking capability. Work is under way to establish function values that span our entire computing complex and workload for Eqs. (1) and (2).

Several lessons learned from our experiences now seem obvious:

1. It is difficult to extrapolate performance predictions from lower to higher benchmark levels because of conflicting factors:
   - overhead is introduced by compilers and operating systems,
   - some users tune their programs better than others,
   - natural evolution in most workloads makes it difficult to be sure that the lower level

benchmarks accurately portray the current workload,
- system software changes affect different benchmarks differently, and
- program initialization and clean-up functions are usually ignored.

2. Users are more interested in their perception of performance than in benchmark results.

3. The answers to questions such as 1-6 discussed earlier are as important as performance to most users.

The following additions to traditional benchmarking may not improve performance, but we believe they will go a long way toward improving user satisfaction and productivity.

- A list of questions similar to 1-7 should be established through mutual agreement of users and systems staff.
- A list of goals related to performance and to the questions should be established in similar fashion.
- A formal process for periodically answering the questions and evaluating how well goals are met should be created.

Two important steps toward addressing Questions 1-7 have been implemented at Los Alamos. A Pearson survey [13] tailored to the needs of Los Alamos users has been performed, analyzed, and documented. Information from this survey and experience gained from conducting the survey will help us in defining better procedures for establishing questions and goals that are important to the users. An extensive performance and workload data collection and analysis effort is underway. Results from these two activities are expected to provide much of the data required to establish good values for the variables of Eqs. (1) and (2).

REFERENCES

1. C. J. Theaker and G. R. Frank, "An Assessment of the MUSS Operating System," Software Practice and Experience, (John Wiley and Sons, 1979) Vol. 9, pp. 657-670.

2. L. E. Nolan and J. C. Strauss, "Workload characterization for Time Sharing System Selection," Software Practice and Experience (John Wiley and Sons, 1974) Vol. 4, pp. 25-41.

3. D. Holdsworth, G. W. Robinson, and M. Wells, "A Multi-Terminal Benchmark," Software Practice and Experience (John Wiley and Sons, 1973) Vol. 3, pp. 43-61.

4. A. R. Gay, "Benchmarking a Multi-Access System," Software Practice and Experience (John Wiley and Sons, 1980) Vol. 10, pp. 45-56.

5. H. J. Curnow and B. A. Wichmann, "A Synthetic Benchmark," The Computer Journal (February 1976) Vol. 19, pp. 43-49.

6. S. Mamrack and J. M. Randall, "An Analysis of a Software Engineering Failure," The Computer Journal (November 1977) Vol. 20 pp. 316-320.

7. K. D. Eason, "Understanding the Naive Computer User," The Computer Journal (February 1976) Vol. 19, pp. 3-7.

8. Y. L. Varol, "Some Remarks on Computer Acquisition," The Computer Journal (May 1976), Vol. 19, pp. 127-131.

9. B. B. Chicklis and R. W. Hanson, "User Considerations Important in the Design of Network Operating Systems," Trends and Applications 1976: Computer Network, IEEE report 76CH1143-7C (1976) pp. 58-62.

10. E. B. James, "The User Interface," The Computer Journal (February 1980) Vol. 23, pp. 25-28.

11. D. Golden, "A Plea for Friendly Software," ACM Software Engineering Notes (October 1980) Vol. 5, pp. 4 5.

12. R. G. Estell, "Benchmarks and Watermarks," Performance Evaluation Review (Fall 1980) Vol. 9, pp. 39-44.

13. S. Pearson, "Measurement of User Satisfaction," Proceedings CMG-X, Dallas, Texas (December 1979) pp. 49-58.

14. D. Deese, "Experience Measuring User Satisfaction," Proceedings CMG-X, Dallas, Texas (December 1979) pp. 59-66.

15. "IPAD Evaluations and Alternatives," Boeing Commercial Airplane Co. report D6-IPAD-70036, Vol. 3 (March 1980).

16. F. P. Brooks, "The Mythical Man Month," Addison-Wesley Publishing Company (1975)

17. J. N. G. Brittan, "Design for a Changing Environment," The Computer Journal (February 1980) Vol. 23, p. 13.

18. R. E. Fulton, "National Meeting to Review IPAD Status and Goals," Astronautics and Aeronautics (July/August 1980) p. 49.

19. A. McKnight, "LASL Benchmark Performance 1978," Los Alamos National Laboratory report LA-7957-MS (August 1979).

20. T. Keller, "CRAY-1 Evaluation, Final Report, Los Alamos National Laboratory report LA-6456-MS (December 1976).

21. T. Keller and D. S. Lindsay, "Development of a Performance Evaluation Methodology for the CRAY-1," Proceedings 1977 SIGMETRICS Conference, Washington, D.C. (November 1977) pp. 169-176.

22. F. Baskett, J. Howard, and J. Montague, "Task Communication in DEMOS," Proc. SIGOPS, Purdue University (November 1977) pp. 23-32.

23. J. Thompson, "Facility for Operator Control and Utilization Statistics (FOCUS) Functional Specification," Computing Division, Los Alamos National Laboratory, internal memorandum, January 1981.

24. I. Bucher, A. Hayes, "I/O Performance Measurement on Cray-1 and CDC 7600 Computers," Proc. CPEUG 80, Orlando, Florida (October 1980).

25. R. Brice and J. W. Anderson, "Performance Evolution in a Large Scale System," Proc. CPEUG 80, Orlando, Florida (October 1980).