A LOAD LEVELING SUPPORT METHODOLOGY FOR NETWORKING*

Stephen R. Kimbleton and Helen M. Wood
National Bureau of Standards

ABSTRACT

Networking provides an effective means for resource sharing; however, its utility for load leveling requires further demonstration. This paper examines obstacles to load leveling and concludes that the key issues relate to the possible loss of control over remotely processed jobs and data. Although an organizational framework necessary to provide appropriate control exists, the technological support mechanism necessary for control in the form of processing guarantees has not yet been developed. Such development requires an essentially real time control capability for determining both the feasibility of accepting remote jobs as well as the guarantees which can be stipulated when such jobs are offered for processing. The key component of this capability is shown to be a very fast analytically driven simulation technique which can be invoked at a given site each time a job is offered for processing. We note in passing that this control capability can be viewed as the "other side" of control limits. That is, while control limits are concerned with determining when system performance has strayed out of bounds, a control capability is concerned with dynamic control of system workload to keep system performance within bounds.

1. AN INTRODUCTION TO LOAD LEVELING

Computer communication networks are often justified in terms of the opportunity which they provide for load leveling and resource sharing. (Load leveling refers to the movement of workload among homogeneous machines to better match global computing resources to global computing resource requirements. Resource sharing refers to the opportunity to access "unique" resources in a heterogeneous network by users not physically colocated with those resources.)

Existing evidence demonstrates that resource sharing works; users are using networking technology to access unique capabilities such as ILLIAC IV, the DATACOMPUTER, and other special purpose capabilities. In addition, papers have appeared demonstrating the economic advantages gained via resource sharing for matching user computing requirements to the capabilities of different systems [ALSBP 75].

There is little published literature to document the organizational acceptability of load leveling used on a continuing basis as a means for peak workload processing. In this paper we argue that this is due to significant organizational and technological obstacles. The remainder of this section surveys some of the most significant of these impediments. Given this summary, the need for a technological supporting mechanism is clear. Section 2 structures the basic nature of the required support and section 3 describes one of the major components in developing a support mechanism. Section 4 provides two examples which, although far from being definitive, do indicate that load leveling does not provide significantly less expensive computing. Section 5 contains concluding remarks and some summary observations.

ORGANIZATIONAL OBSTACLES TO LOAD LEVELING

There are three primary organizational obstacles to load leveling: lack of required supporting information; uncertain economic implications; and loss of organizational control.

Information Support Requirements

Efficient remote access to computer systems requires a significant amount of supporting information in the form of the resources

*This paper is a contribution of the National Bureau of Standards and as such is not subject to copyright.

existing at a given location, their availability, requirements for their access, and the detailed technical information needed to support access given that it is permitted. In addition, some sort of counseling function must be provided to aid the first time user. (Since networking promises to significantly expand the spectrum of users of a given system, increased consulting costs constitute a real cost which must be recovered.)

Although comprehensive centralized information support capabilities do not apparently exist at this time, there is significant evidence that such a capability could exist [BENOJ 74]. Indeed, the Network Information Center at the Stanford Research Institute [NIC 75] provided some classes of supporting information and organized and distributed an ARPANET resources handbook for a significant period of time. There are few technological impediments to expanding and placing online such an information support base. Thus, we conclude that the first organizational obstacle can be eliminated by proper application of existing technology.

Economic Implications

Procurement of computing resources from vendors costs real dollars; payment for utilization of these resources within an organization is usually via some combination of real money and "funny" money. Since the amount of real money tends to be quite limited, most managers of computing installations are extremely reluctant to see real dollars flow across organizational boundaries [KIMBS 76A]. Thus, the balance of payments/flow of funds issue in remote utilization of resources is significant. This issue is one of the several major organizational factors being considered in a major study currently underway [NIELN 76]. It is interesting to note that although most organizations are willing to accept funds supporting remote utilization of their computational resources, few appear particularly enthusiastic about paying for utilization of other remote resources.

The potential for a free market in computing services supported by networking has significant implications which are difficult to predict. However, it is clear that the minimum requirement for effective free market operation is a network wide accounting capability. (Note that bilateral agreements could be utilized without requiring the existence of a network wide accounting capability. However, such agreements are organizationally cumbersome, require separate account establishment for each new participating organization, and, if there are N participating hosts, require N**2 rather than N billing operations.) It is also clear that there are no significant technological obstacles to the implementation of such a capability and that the necessary costing technology is essentially at hand [KIMBS 76A], [TMPCS 76]. Thus, from a technological point of view economic obstacles to load leveling seem removable; the organizational implications and acceptability of doing so remain unproven.

Control

The positive benefits of resource sharing and load leveling which can be achieved via networking must be balanced against the detriments implicit in loss of managerial control. Such loss of control is reflected in loss of absolute control over data access coupled with the possibility of losing access to a system or being provided only a limited fraction of what were thought to be guaranteed resources when the "crunch" comes.

Loss of organizational control over data and systems has significant data security and integrity implications. The current state of the art in security implementations does not yet provide absolute data security; however, progress is being made in such areas as data encryption [NBS 75], ADP physical security and risk management [NBS 74], and personal authentication techniques [COTTI 75]. These emergent capabilities promise a significant enhancement in installation security. Having noted the importance of acceptable security as a prerequisite to load leveling, we now consider additional requirements without further discussion of security issues.

Assuming security capabilities are deemed acceptable, network utilization for peak workload processing has two key requirements: assurance that processing capacity will exist when needed and guaranteeing that a job which is to be processed remotely is indeed completed within the agreed upon response/turnaround time requirements. The first requirement simply states that the global network processing capacity must exceed the global network processing requirements for at least the critical types of jobs. We assume that this condition is satisfied and our concern rests on determining the potential for satisfying the second requirement.

Provision of job processing guarantees by a remote site requires development of appropriate managerial and technical supporting capabilities. The next section

structures these requirements in some detail.

2. LOAD LEVELING REQUIREMENTS

Assuring that a remote site will, indeed, process a job within the anticipated/agreed upon interval implies two requirements. The first is that the site be able to properly evaluate the processing requirements of existing jobs; the second is that its management be able to guarantee that the normal processing priorities used in developing these estimates are not arbitrarily altered.

ORGANIZATIONAL ASSURANCES

The "squeaky wheel" syndrome is well understood in data processing circles. When the computing crunch comes, it is usually the squeakiest (most vocal) among the various groups competing for service (within a given priority class, at least) who gets the "best" service. Squeaking is best accomplished via proximity; thus, the remote user of computing services is in an intrinsically disadvantageous position.

In view of this fact, effective load leveling requires development of a managerial framework to ensure equality among local and remote users of a given computing resource. The Wholesale/Retail approach to management of computing resources has been developed to provide such a framework [STEFE 73,76].

To the extent to which the squeaky wheel syndrome is applicable, management will tend to avoid load leveling because of the possible failure to receive guaranteed service during peak processing requirement times. Thus, any realistic load leveling agreement between facilities requires absolute assurance that each accessor to a given facility will receive the agreed upon level of service. Fortunately, there is ample evidence to indicate that the giving of such guarantees is feasible and organizationally supportable provided that the priorities of all accessors to a given system are rigidly enforced [WILLL 76].

TECHNOLOGICAL REQUIREMENTS FOR LOAD LEVELING

Utilization of load leveling requires implementation of the appropriate network level primitives to support movement of jobs and data among systems and to permit remote initiation of jobs at the site at which they are to be executed. Since load leveling occurs among homogeneous systems which, functionally, may be viewed as object code compatible systems, the file transfer capabilities available within, for example, the ARPANET are sufficient to support the movement of object modules and entire data files. (It should be noted that at the present time, capabilities for remote record access are not generally available [KIMBS 76B].) Further, remote initiation of jobs is facilitated by the presumed similarity among (operating) systems implied by object code compatibility. Thus, we assume that the basic technology required to support movement of object modules and files of data exists; we now consider the supporting requirements implicit in effective utilization of this technology.

Given the discussion in section 2, it follows that the primary technological problem in supporting load leveling is development of an appropriate remote processing control capability. Such a control capability has three components: i) determination by an individual host of when to resource out jobs onto the network; ii) identification of that host to which jobs are to be transmitted given that a decision has been made to resource out some jobs; and iii) establishment of a methodology for providing processing guarantees by the site to which the jobs have been transmitted.

Note that the first two components are required by the site which is seeking to resource out a portion of its workload while the third is required by sites which are candidate acceptors of such workload. The first component can be approached either pragmatically or in terms of developing an explicit mathematical policy. It seems reasonable to assume that satisfactory pragmatics can be developed and will suffice initially; development of policies for determining when to resource out seems to be a likely area for future research.

Selection of a remote site for workload processing from among a collection of candidate sites is an area of increasing concern. Presently, measurement techniques of both a static and dynamic nature have been considered, and the initial evidence indicates that semiautomated site selection is feasible with a limited amount of overhead [MAMRS 76A,B,C].

Although the need for processing quarantees by the accepting site has been recognized [STEFE 76], a methodology for their provision does not yet appear to exist. It is to this issue that we now turn.

3. AN APPROACH TO PROVIDING PROCESSING GUARANTEES

Provision of processing guarantees breaks down into two categories: i) provision of response time guarantees for interactive jobs, and ii) provision of turnaround time guarantees for batch jobs. In addition, since a given system will normally process both types of jobs, the fraction of resources for each category must be controlled.

Implementation of a control limit approach to provide response time guarantees for interactive jobs is relatively straightforward because of the real time nature of such jobs. Thus, if response time is unacceptable one would avoid accepting any new interactive jobs and would also seek to either lower the priority or halt the execution of any jobs in the interactive mix which are deferrable. Moreover, the device management algorithms for processors, primary memory, and secondary storage devices can be tuned to discriminate among those interactive jobs which are more intensive users of system resources.

Provision of turnaround time guarantees for batch jobs is significantly more difficult since the effects of a decision to accept/reject an offered job as well as the determination of a reasonable processing time guarantee will not be known until the completion of job processing time. Identification of the requirements implicit in the provision of suitable guarantees requires consideration of the major sources of delay for a batch job and identification of the options for controlling the maximum delay corresponding to each of these sources.

The turnaround time (TAT) for a batch job may be represented as:

$$TAT = JTT + SRT + JRT$$

where JTT is the time required to transmit the program and data to the processing location, SRT is the system residence time of the job and JRT is the time to return the program and data to the location from which it came. Note that because programs are usually smaller than the files they access, the program code may be prepositioned to further reduce JTT and JRT. Prepositioning data code is usually undesirable because of the multiple update problem.

Computation of JTT and JRT is straightforward given knowledge of the amount of information to be transmitted. However, in cases where file size is large, the available bandwidth will have a strong inhibiting effect on the reasonable size of a file which may be transmitted. To the extent that this is true, this is a key factor that may inhibit load leveling. In any case, empirical measurement provides reasonable estimators for JTT and JRT. Further, knowledge of the range of observed transmission rates between source and sink permits identification of reasonable upper and lower bounds for the transmission rate and, consequently, the delay.

Estimation of the system residence time of a job requires estimation of both the queuing time QT and the processing time PT of the job. Clearly, both QT and PT are influenced by the external scheduling requirements of the system. Typically, one estimates a few averages and uses the result to estimate SRT. However, since the remote host must, in effect, provide service guarantees, such a crude estimation approach is not feasible. Indeed, in a networking environment, this approach becomes even less feasible since systems which are net providers of computing resources may be unable to adequately estimate the total processing requirements imposed on them.

The alternative to static estimation is dynamic estimation of the SRT. Such estimation should permit dynamic determination for any offered job of the processing guarantees which can be provided. Such guarantees are clearly a function of previous guarantees which have also been provided. Moreover, they apply to the continuous interval of time between the time at which a job is offered and the time of and completion of the last job in the queue. In particular, the option should be available to accept a job with a very tight turnaround time guarantee if the processing guarantees provided to all jobs currently in the queue permit insertion of this job ahead of the other jobs and still permit all guarantees to be met.

It follows from the proceeding argument that the provision of appropriate processing guarantees requires a real time turnaround time estimation capability. This capability can be viewed as the "other side" of a control limit capability. That is, the purpose of control limits is to determine when system performance is out of bounds for a given installation, while the purpose of the control capability is to adjust the workload to keep system performance in bounds. Note that the operational cost of this capability must be sufficiently low to permit its repeated invocation whenever a new job is offered for service whose processing time guarantees are not trivially satisfied. It seems likely that such jobs will arrive frequently in a networking environment.

REAL TIME SRT ESTIMATION

Real time SRT estimation requires estimation of both the scheduling delay QT and the processing delay PT. Such estimates must explicitly reflect the effects of scheduling constraints imposed by: availability of serially reusable resources; deadlines; precedence relations;

and organizational priorities. The utilization of a heuristic scheduler to investigate these factors has been discussed in [KIMBS 74]. This scheduler is based upon utilization of a very fast performance prediction technique which will now be described.

Estimation of the performance of a system is traditionally approached via either analytic or simulation techniques. The traditional disadvantage of analytic approaches is the relative lack of detail which they provide and the inability to represent system characteristics with any precision; their advantage is usually speed of computation. Simulation approaches, in contrast, are capable of providing relatively high accuracy together with an arbitrarily precise degree of fidelity in representing system characteristics, both hardware and software. The disadvantage is reflected in their speed of computation which, for a simulation providing information on device delays and utilizations, is typically on the order of 1-10 times slower than real time. Clearly, the cost of a pure simulation approach is totally unsuited to support of a real time control capability for a computer system.

As long as one restricts oneself to pure analytic or simulation techniques, it seems likely this impasse will continue. However, it is reasonable to investigate the feasibility of a mixed or hybrid approach which combines simulation and analytic techniques to achieve the desired level of speed while avoiding the sacrifice of too much precision. Such an approach has been described in [KIMBS 75]. A more detailed validation of the approach for a PDP 10 computer system is included in [WOODH 76]. In effect it decomposes the problem of performance prediction into two components: prediction of the performance of the system during those periods of time when the composition of the mix is constant, and aggregation of the resulting statistics to yield overall shift or job performance. It develops that this approach yields the required speed since analytic techniques can be used for the first part. Moreover, it yields a reasonable level of accuracy since the traditional statistical summarization approaches common to simulation are used to obtain the aggregate statistics. (It should be noted, however, that development of the analytic model requires some care to keep the computational complexity linear in terms of the number of devices in the system. In particular, approximation techniques are required.) The result is a capability for computing system performance at a processing cost of approximately 1 second/job on a PDP-10 TENEX system. Further refinements may be able to significantly reduce even this number.

Load leveling is clearly desirable for balancing workload and ensuring more acceptable system performance across the collection of computer systems within a computer communication network. It might also be conjectured that load leveling would lead to significantly reduced aggregate system costs in processing jobs. Using the performance prediction technique discussed in the last section we have developed a few examples which partially investigate this hypothesis.

4. TWO EXAMPLES

To determine economies achievable via load leveling, a series of examples were run. Two of the most illustrative are reported here. Note that the data in these examples was obtained via the analytically driven simulator discussed above rather than via direct execution on some object system. This was done since the former approach supported costing of the jobs on the basis of resources utilized rather than on the basis of information returned by the accounting algorithm which, as noted in [KIMBS 76A], is usually biased to perform a resource allocation function in addition to direct resource costing.

SYSTEM CHARACTERISTICS: -

NO. OF DRUMS =	0
NO. OF DISK DRIVES =	3
NO. OF TAPE DRIVES =	0
NO. OF NON-SHARED DISK DRIVES =	0
AMOUNT OF CORE MEMORY =	100
ROTATION TIME OF DISK =	16.6667
NUMBER OF BLOCKS PER TRACK =	5
TOTAL NUMBER OF CYLINDERS =	411
MINIMUM SEEK TIME =	7.0000
MAXIMUM SEEK TIME =	50.0000
AVERAGE SEEK TIME =	28.0000
COST PER UNIT TIME OF CPU =	10.000000
COST OF CORE PER UNIT TIME =	1.000000
COST OF DRUM PER UNIT TIME =	0.015000
COST OF DISK PER UNIT TIME =	0.006800
COST OF TAPE PER UNIT TIME =	0.005000

TABLE 1

The jobs to be executed on the system described in Table 1 are characterized via a Job/System Description (JSD) identified in [KIMBS 75]. In essence this characterization consists of an endogenous and an exogenous portion. The exogenous portion is concerned with describing the environment in which the job is to execute and the endogenous portion describes the resource demands made in this environment. Since the examples involve comparisons of system performance for two jobs being concurrently executed, the exogenous component of the JSD which includes such factors as priorities,

precedence relations, etc., is uninteresting. We now describe the endogenous components for each of the two examples.

EXAMPLE 1. This example is concerned with disk contention in a three disk system where the disks are labelled 1, 2 and 3. We denote by J(i) a job accessing only disk i and by J(i,k) a job accessing disks i and k. All jobs have the same active interval of 10 ms. (An active interval is the total amount of processor time required between the completion of one I/O interval and the initiation of the next). Three runs were made corresponding to the concurrently executing pairs: [J(1),J(2)], [J(1,2),J(3)], and [J(1,2),J(2,3)]. In each case the number of I/O operations per device was 500. and the amount of data transferred per I/O operation was one page.

Let us denote the average CPU utilization by UCPU, the System Reward Function by SRF and the length of the shift or simulated time by ST. (Note that the System Reward Function is a dollar weighted device utilization over the shift [KIMBS 74].) These triples for the three runs were: (15.0, .1545, 68), (15.4, .1587, 66) and (15.4, .1587, 66). Since the total resource utilization costs for executing the jobs processed during a shift is simply the product SRF*ST, it follows that these costs for the three runs are 10.5, 10.5, and 10.5.

Given the nonexistent differences in Example 1, reflection seemed to indicate that the marginal costing differences might really just reflect the fact tha I/O costs are only a small fraction of total system costs as Table 1 illustrates. This led to the second example.

EXAMPLE 2. This example studied the effects of processor contention. Two job types existed: Type A jobs had 100 ms. active intervals while Type B jobs had 10 ms. active intervals. Three runs corresponding to the job types [A,A], [A,B] and [B,B] were made. I/O accesses were to different devices so no I/O contention was present. With the notation above, the results were: (67.3, .6286, 22), (15.0, .1545, 68) and (21.3, .2061, 55). Again, upon forming the product SRF*ST the results were: 13.8, 10.5 and 11.3. Assuming two processors and looking at the assignments of two type A jobs to one and two type B to another versus a mixed assignment of one type A and one type B job to each processor, the total reward (cost) over both processors is 25.1 versus 21.0. While this is a savings of 19.5%, it is less than breathtaking even though the jobs are remarkably dissimilar in terms of their processor requirements.

5. CONCLUDING REMARKS

We conclude that load leveling is a feasible utilization of networking technology, that its effective utilization requires both organizational and technological support and that the major dimensions of required support can be implemented within the available technology. Perhaps the major remaining obstacle to adequate utilization of networking for load leveling is the need for appropriate Network Operating System like capabilities [KIMBS 76B]. Given its existence, utilization of a load leveling technology is likely to prove highly satisfactory to the individual user in terms of increased system responsiveness. Although more exhaustive studies may be able to identify cost savings realizable through load leveling, our preliminary findings do not indicate significantly reduced costs. This should be contrasted with the costing implications of resource sharing to better fit processing requirements to machine capabilities and the nuances of the costing algorithm as reported in [ALSBP 75].

REFERENCES

ALSBP 75 Alsberg, P.A. "Distributed processing on the ARPA network -- measurements of the cost and performance tradeoffs for numerical tasks," Proceedings Eighth Hawaii International Conference on System Sciences, Univ. of Hawaii, Honolulu, Hawaii, Jan. 1975, 91-94.

BENOJ 74 Benoit, John W., and Erika Graf-Webster, "Evolution of network user services - the network resource manager", Proceedings of the 1974 Symposium on Computer Networks: Trends and Applications, IEEE Computer Society, May 74, 21-29.

COTTI 75 Cotton, Ira W., and Paul Meissner, "Approaches to controlling personal access to computer terminals," Proceedings of the 1975 Symposium on Computer Networks: Trends and Applications, IEEE Computer Society, June 1975, 32-39.

KIMBS 74 Kimbleton, Stephen R., "Batch computer scheduling: a heuristically motivated approach," Proceedings of the Second Annual SIGMETRICS Symposium on Measurement and Evaluation, Montreal, Canada, Sep. 30 - Oct. 2, 1974, 189-198.

KIMBS 75 ----------, "A heuristic approach to computer systems performance improvement, I: a fast performance prediction tool", Proceedings of the

AFIPS 1975 National Computer Conference, Vol. 44, AFIPS Press, Montvale, N.J., 1975, 839-846.

KIMBS 76A ----------, "Considerations in pricing distributed computing", Proceedings of the SIGMETRICS Technical Meeting on Pricing Computer Services, 5C:1, Mar. 1976, 22-30.

KIMBS 76B ----------, and Richard L. Mandell, "A perspective on network operating systems", Proceedings of the AFIPS 1976 National Computer Conference, Vol. 45, AFIPS Press, Montvale, N.J., 1976, 551-560.

MAMRS 76A Mamrak, Sandra A., "A network resource sharing module to augment user cost-benefit analysis," Proceedings 1976 Symposium on Computer Networks: Trends and Applications, IEEE Computer Society, Nov. 1976.

MAMRS 76B ----------, "A predictive response time monitor for computer networks," Proceedings of the Third International Conference on computer communication, Toronto, Canada, Aug. 1976, 626-630.

MAMRS 76C ----------, "Response time prediction as a network resource sharing service," to appear, 1976.

NBS 74 National Bureau of Standards. "Guidelines for automatic data processing physical security and risk management," FIPS PUB 31, June 1974.

NBS 75 National Bureau of Standards. "Proposed standard encryption algorithm for computer data protection." Federal Register, 40:52 (August 1, 1975), 12134-12140.

NIC 75 ARPANET Resource Handbook, Network Information Center, Stanford Research Institute, Sep. 1975.

NIELN 76 Nielsen, Norman R., "Simulation of institutional behavior in a national networking environment", Proceedings of the 1976 Winter Simulation Conference, National Bureau of Standards, Dec. 1976.

STEFE 76 Stefferud, Einar, Text of talk given at SIGMETRICS Technical Meeting on Pricing Computer Services, Proceedings of SIGMETRICS Technical Meeting on Pricing Computer Services, March 1976, 5C:1, 31-70.

STEFE 73 Stefferud, E.: Grobstein, D.L.: and Uhlig, R., "Wholesale/retail specialization in resource sharing networks," IEEE Computer, 6:8, August 1973, 31-37.

TMPCS 76 Proceedings of the SIGMETRICS Technical Meeting on Pricing Computer Services, 5C:1, Mar. 1976.

WILLL 76 Williams, Leland H., "Network computing", IFIP-INFOPOL, Mar. 1976.

WOODH 76 Wood, Helen M., and Stephen R. Kimbleton, "Validation of ASIM: a performance prediction tool", to appear, 1976.