Load Leveling for Control of Distributed Processing Systems

H. W. Tyrer, Ph.D., G. Ramakrishnan
Department of Electrical and Computer Engineering
and the Engineering Computer Network
University of Missouri
Columbia, Missouri 65211

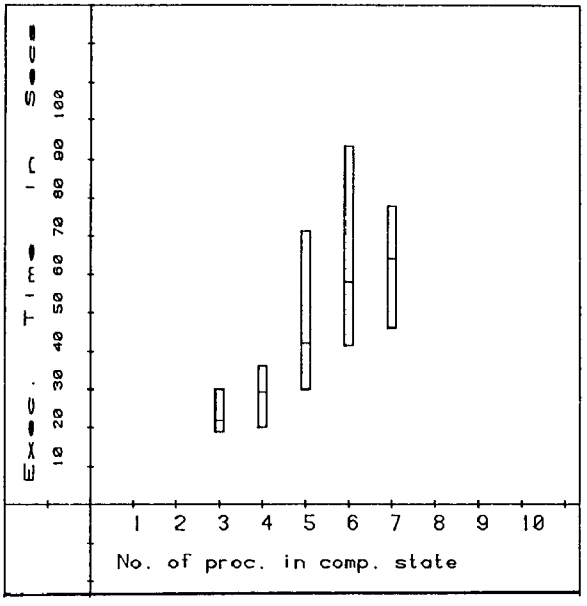
Equalization of computing loads among distributed processors can be the basis for a decentralized operating system. Such equalization can be stated as a design principle: each node acts in its own best interest. A node's best interest is to maintain a high rate of process execution time.

For a process transferred among nodes, we assess whether the incurred overhead cost in transferring the task will reduce the load to make such transference worthwhile. A second problem is the existence of such thresholds.

Methods

We are developing the load leveling system on a Digital Equipment Corporation based ethernet system. Our two experimental machines are: a VAX 11/780 with 16 Mbytes of main memory and 860 bytes of secondary storage; and a VAX 8800 with 32 Mbytes of main memory and 1200 Mbytes of secondary storage. The two machines communicate via an ETHERNET system supported by DECNET software.

By using task-to-task communication technique, a large program was executed in the remote machine (VAX 8800) simultaneously. The parameters measured included the following: execution time on VAX 11/780, time for copying executing and deleting the image file from VAX 11/780 to VAX 8800.



Results

Performance data of an 11/780 connected to an 8800 can demonstrate the interaction of a slower node with a faster node (8800). The table shows that the average execution time of the VAX 11/780 is between two and three times that of the execution time, copying time and deletion time of the 8800 in seconds. The 11/780, running with three computable processes is compared to the 8800 which can be kept running with six computable processes, also the converse, is useful.

Ready Processes	11/780 avg. exec. time	8800 avg. exec. transfer time
3	22	9
4	29	12
5	42	13.6
6	58	16.6
7	64	23.2

Average CPU Utilization: 98%
Average Page Fault Rate: 55
Times in seconds.

We now define thresholds to separate the three regions of operation. The figure displays the distribution of execution time as a function of the number of processes in the computable (ready) state. Region one is identified as three or less processes in the computable state. Here an additional process does not significantly increase the expected execution time. Region two could be identified as four processes in the computable state. Here an additional process could substantially (by a factor of two and one half) increase the execution time. Region three corresponds to five or more processes in the computable state and the node will transfer a task if it is allowed.

Conclusions

Clearly, we have demonstrated that it is worthwhile to transfer the task to reduce the load on a node. Furthermore, parameters can be identified to create thresholds to trigger a node to act in its own best interest.