Proceedings of the 1989 Winter Simulation Conference
E.A. MacNair, K.J. Musselman, P. Heidelberger (eds.)

Load Sharing in Hierarchical Distributed Systems

S. A. Banawan
Dept. of Computer Science
University of Houston
Houston, TX 77204

J. Zahorjan
Dept. of Computer Science
University of Washington
Seattle, WA 98195

Abstract

Load sharing has been the focus of a great deal of research as a means of enhancing the performance of distributed systems. In this paper we evaluate the potential benefits of load sharing in *hierarchical* distributed systems. A hierarchical system consists of sub-systems, each of which is a distributed system in its own right. Our goal is to determine the level(s) at which load sharing may be exercised to improve the system performance at minimum overhead. Our simulation results indicate that most of the potential benefits can be achieved by employing load sharing locally. The additional benefit gained from global load sharing is minimal and does not justify the cost associated with it.

1 Introduction

In recent years, dynamic load sharing has been the focus of a great deal of research as a means of enhancing the performance of distributed systems[LM82][SS84][WM85][ELZ86][ZF87]. Most of the previous work, however, considered *single-level* systems. In this paper we study the performance of load sharing in hierarchical systems. Our goal is to determine the appropriate level(s) at which load sharing may be exercised to improve the system performance at minimum overhead.

Although there are many ways to connect individual nodes into a hierarchical system, we will consider a particular class of these systems, namely those with a two-level hierarchy. A two-level hierarchical system can result from merging a number of smaller subsystems. For instance, consider a campus-wide distributed system. The machines within each department might form sub-systems or clusters of nodes. At a higher level clusters are connected to facilitate interdepartmental sharing and communication. Among the virtues of a two-level hierarchy is the ability to accommodate a growing number of nodes in a systematic way. In addition, it naturally reflects:

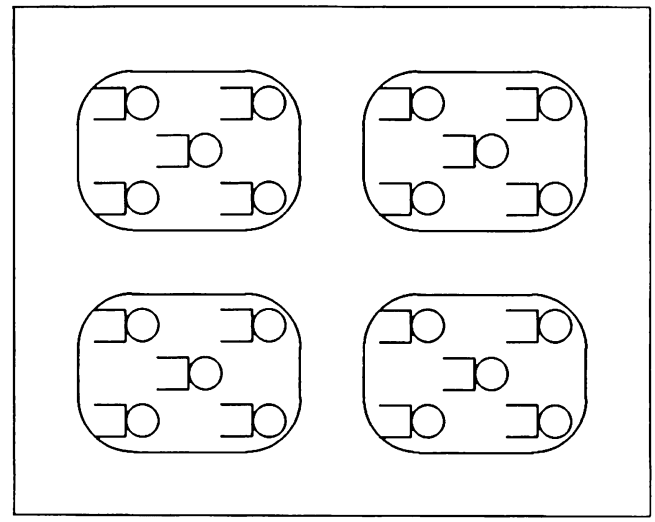

Figure 1: A cluster-based system

- physical proximity (e.g., a cluster may consist of machines that exist in the same building)
- administrative boundaries (e.g., a cluster may consist of machines that belong to the same department)
- other groupings (e.g., a cluster may consist of machines that share the same file server)

Figure 1 shows an example of a cluster-based system. It consists of four clusters, each of which has five nodes.

We assume that each cluster consists of a medium number of homogeneous nodes that communicate over a local area network. Such a network guarantees a uniform distance between any pair of nodes in the same cluster. That is the cost of transferring

Table 1: Load Sharing in Hierarchical Systems

Policy	*Load Sharing* High level	Low level
1. None	No	No
2. Inter-cluster	Yes	No
3. Intra-cluster	No	Yes
4. Both	Yes	Yes

a job between any two nodes is the same. Furthermore, this cost is typically negligible in network of workstations which share the same file servers. Consequently, performing load sharing in such an environment has minimal overhead as was demonstrated by Korry [Kor86]. Note, however, that the uniform distance assumption may not hold if the nodes belong to two different clusters. While we do not specify the interconnection network over which the communication takes place between nodes in different clusters, we require that the communication cost is at least as expensive as the communication cost between nodes in the same cluster[1].

In a hierarchical distributed system, load sharing can be performed locally, i.e., within each cluster, globally, i.e., across clusters, or both. Thus, there are four different cases as given in the following Table 1

We are interested in evaluating the performance of load sharing in each case. Such an evaluation can be used to choose the appropriate level at which load sharing may be exercised to realize the potential benefits at minimum cost. Our evaluation is based on simulation since analytic techniques will be prohibitively expensive due to the combinatorial explosion of the state space, even in systems with a small number of clusters that have a few nodes.

This paper has the following organization. In §2 we discuss the performance of inter-cluster or global load sharing. §3 presents the case of both global as well as local load sharing. The performance of local load sharing alone (i.e., within a cluster) has been investigated by several researchers, e.g., see [ZF87] and [Ban87]. A comparison between the potential benefits from global, local, both global and local, and no load sharing is given in §4. Finally, §5 summarizes our conclusions.

[1]The communication cost between clusters may be much higher if messages are routed via a gateway, if less bandwidth is allocated for sending messages between clusters, etc.

2 Inter-cluster Load Sharing

By *inter-cluster* load sharing we mean that a job originating at one cluster may be transferred to another cluster for execution. Within a cluster, no effort is made at load sharing. Once a cluster is chosen, the job is arbitrarily assigned to some node in this cluster and it runs there until completion. In other words, we consider only non-preemptive load sharing. Eager *et al.* [ELZ88] have demonstrated that it is unlikely that preemptive policies can yield performance that is significantly better than their non-preemptive counterparts.

Note that by evaluating the performance of inter-cluster load sharing, we do not suggest that it is a practical approach to load sharing, rather our goal is to highlight the potential benefits of load sharing among a collection of subnetworks or clusters. Clearly, cluster selection in this case should be based on a criterion that optimizes our performance measure, mean response time.

We assume that the system is homogeneous. Each cluster has the same number of nodes and each node has the same service rate. One obvious criterion for cluster selection, given the homogeneity of the system, is to assign each new job to the cluster that has the least number of jobs at the job arrival time. In other words, each new job is assigned to the least loaded cluster, where the cluster load is determined by the total number of jobs in that cluster. This can be regarded as a generalization of "join the shortest queue" policy where the "cluster queue length" is actually the sum of the queue lengths at all nodes in the cluster. Although this intuitive measure yields a substantial performance improvement when the cluster size is one, i.e., each cluster is an individual node (see [Ban87]), it may not be very useful if the cluster size is larger.

Since there are many ways to partition the nodes in a hierarchical system, our evaluation of inter-cluster load sharing will be based on two specific cases. In the first case we assume that the cluster size is constant and we vary the number of clusters. In the second case we assume the system has a fixed number of clusters, but the number of nodes per cluster may vary.

Figure 2 shows the mean response time versus system utilization obtained from simulating inter-cluster load sharing. Different curves correspond to systems with different number of clusters. The cluster size, however, is the same in all systems. Each cluster has 10 nodes. Each node is modeled as a single queueing center. Both the interarrival time and the service time have exponential distributions. The load sharing

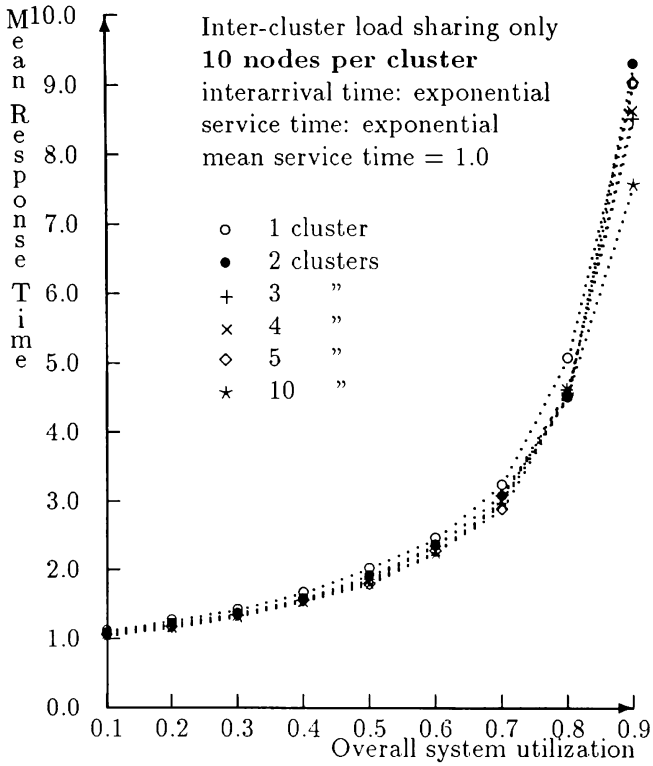

Figure 2: Inter-cluster load sharing – variable number of clusters

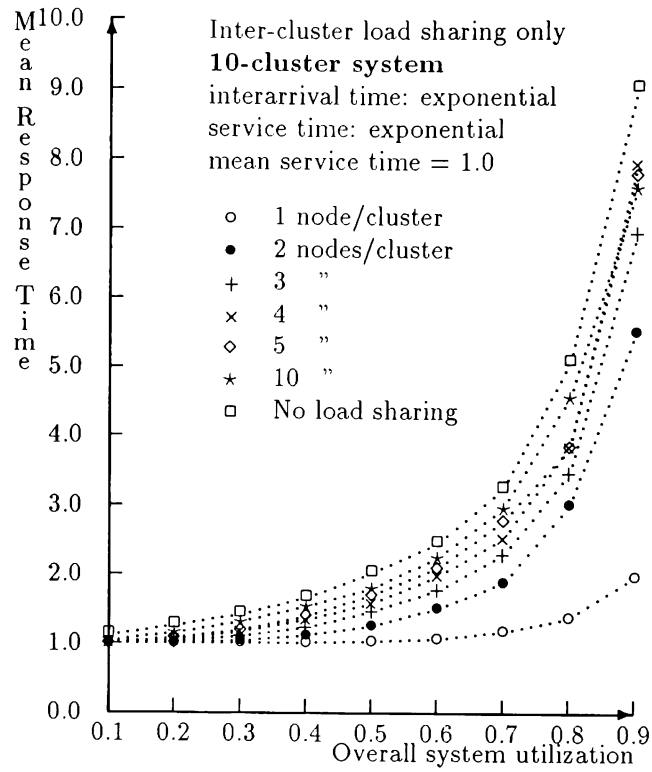

Figure 3: Inter-cluster load sharing – variable number of nodes per cluster

algorithm is invoked whenever a new job arrives. It selects the cluster with the least number of jobs. One of the nodes in that cluster is then chosen arbitrarily to execute the new job. The curve corresponding to a system that has a single cluster gives the performance of the no load sharing case since no job transfer occurs in this system.

Intuitively, the more clusters the better inter-cluster load sharing would perform, because with more clusters in the system we expect the probability of finding a lightly loaded cluster each time a new job arrives to be higher. We know that this is true when the cluster size is one [LM82]. The results shown in Figure 2 indicate that this intuition does not apply to multiple-node clusters. Increasing the number of clusters in the system has a minimal effect on the mean response time of all jobs. In other words, the performance of inter-cluster load sharing in a system with several clusters is very close to that of a system with a single cluster where no load sharing is performed. Note that these results were obtained under the assumption that the cost of transferring a job between nodes in different clusters is the same as if the nodes were in the same cluster. Should job transfer between clusters incur an extra overhead the performance gain from inter-cluster load sharing can become even smaller.

Figure 3 shows the performance of inter-cluster load sharing as we keep the number of clusters constant and vary the cluster size. We assume that the system has 10 clusters. Again, the interarrival time and the service time are assumed to be exponential. The figure shows that a substantial performance gain is possible only when the cluster size is very small. As the number of nodes per cluster increases the benefits of inter-cluster load sharing become smaller and the performance of the system approaches the no load sharing case.

The small performance gain obtained by employing inter-cluster load sharing can be attributed to the fact that clusters are compared using the cluster "macro" state which ignores the distribution of jobs among the nodes in the cluster. Although we select the cluster that has the least number of jobs, it is still probable that some nodes in this cluster may be relatively congested. Unless care is exercised in choosing which node to use, the new job may be assigned to one of these congested nodes which negates the benefits of choosing the best cluster.

3 Inter and Intra-cluster Load Sharing

In this section we are interested in the performance of hierarchical systems when load sharing is performed both globally and locally. Global load sharing alone, or inter-cluster load sharing, which assigns new jobs

to clusters based solely on their macro states, was shown to have little impact on system performance in the previous section.

To perform load sharing both locally and globally we not only have to select the best cluster, but we must also perform load sharing locally, i.e., among the nodes in the same cluster. The problem of load sharing in a single cluster has been addressed in [Ban87]. It has been shown that the potential benefits of local load sharing can be realized by employing a policy that assigns each new job to the node with the shortest queue. In fact, Winston[Win77] has proven that this policy is optimal when both interarrival and service time have exponential distributions. We call load sharing that is performed locally *intra-cluster* load sharing. Thus, to perform both inter- and intra-cluster load sharing simultaneously, the load sharing algorithm must do the following whenever a new job arrives:

1. it finds the cluster c with the smallest *total* number of jobs.

2. it finds the node n in c that has the *shortest* queue length.

3. it assigns the new job to run on node n only.

Similar to the treatment in the previous section, our evaluation of inter- and intra-cluster load sharing will be based on two cases. In the first case we assume constant cluster size and we vary the number of clusters. In the second case we assume the number of clusters is constant as we change the number of nodes per cluster. Figure 4 shows the simulation results in the first case. The cluster size is 10. Both the interarrival time and the service time have exponential distributions. Also, we assume that job transfer between nodes in different clusters does not incur an extra overhead compared with job transfer between nodes in the same cluster.

Figure 4 shows a tremendous performance improvement is possible by implementing both inter and intra-cluster load sharing compared with the no load sharing case. The improvement, however, is less dependent on the number of clusters in the system. In fact, it is hard to distinguish curves that correspond to different number of clusters except at very high utilizations. This result can be explained as follows. Although the cluster size is medium, the number of nodes in each cluster, 10, is still large enough so that each cluster has, most of the time, one or more idle or lightly loaded nodes [LM82]. Thus, it is usually possible to assign each new job to a lightly loaded node in the job's cluster. In other words, the best node in the *best cluster* is almost as good as the best node in *any cluster*.

The simulation results in the second case is shown in Figure 5. The number of clusters in each system is 10. The number of nodes per cluster varies from 1 to 10. Again, a substantial improvement is possible through inter- and intra-cluster load sharing. Consider, for example, the 10-cluster system where the cluster size is 1. Performing both inter-cluster and intra-cluster load sharing in this system is equivalent to performing inter-cluster load sharing only. Since each cluster has a single node, load sharing can greatly improve the system performance (see Figure 2). As the number of nodes increases we get even better performance. This is due to the fact that not only do we select the best cluster but we also choose the best node in that cluster.

4 A Comparison of Load Sharing Approaches in Hierarchical Systems

As we mentioned earlier, in a cluster-based system load sharing may be performed locally, globally or both. Each of these policies has been studied separately. This section puts in perspective the performance results obtained by simulating the same system under all of the above policies. Since we cannot exhaust all possible combinations of configuring a hierarchical system, we consider a few representative cases.

In the first case, we assume that the system has 5 clusters, each of which has 5 nodes. Figure 6 shows the mean response time of all jobs versus system utilization under the four load sharing policies mentioned above. All nodes in the system have the same utilization which is taken as the system utilization. As before, both the interarrival time and the service time have exponential distributions. Also, the simulation assumes that the communication cost is negligible. Thus, executing a job remotely on a node that belongs to a different cluster or to the same cluster does not incur an extra overhead compared with local execution. Obviously, this may not be true in some hierarchical systems. However, assuming otherwise, i.e., the transfer cost between clusters is more expensive, does not change our conclusion, rather it makes it even stronger since our thesis is that little performance gain can be obtained by transferring jobs between clusters.

In the second case we double the number of clusters so that the system has 10 clusters, but the cluster size

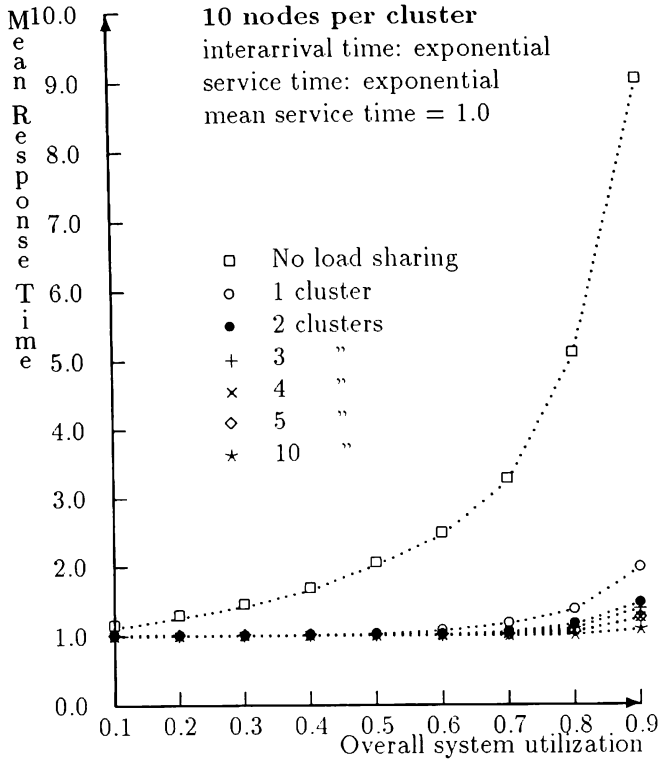

Figure 4: Inter- and intra-cluster load sharing – variable number of clusters

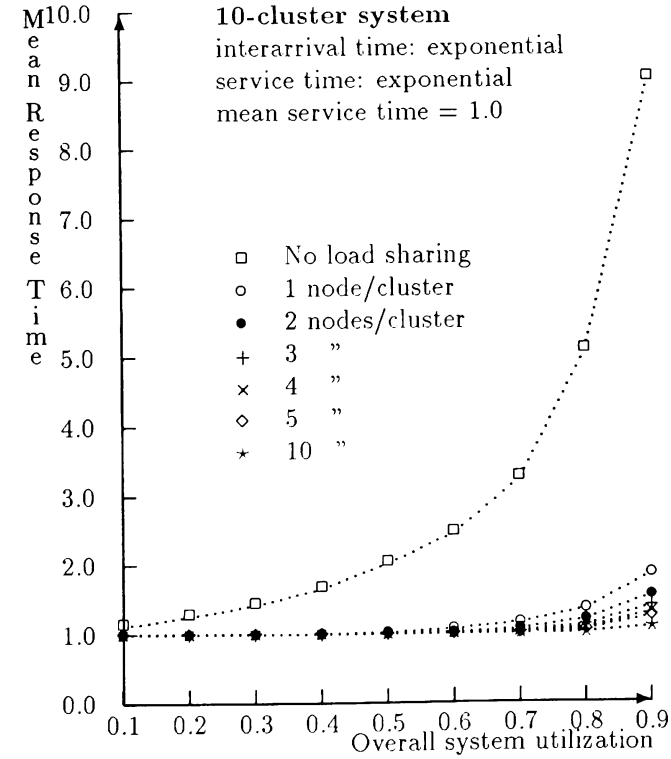

Figure 5: Inter- and intra-cluster load sharing – variable number of nodes per cluster

is still 5. In the third case, we double the cluster size but the number of cluster remains the same, i.e., we have a system that has 5 clusters but the cluster size is 10. Finally, we double both the number of clusters and the number of nodes per cluster. The resultant system has 10 clusters, each of which has 10 nodes. The simulation results corresponding to the last three cases are shown in Figures 7, 8 and 9 respectively.

Several observation regarding the performance curves in these figures are in order:

- In general, load sharing yields a performance that is better than no load sharing regardless of the level at which load sharing is performed.
- Performing both inter-cluster and intra-cluster load sharing is better than performing either one alone.
- The performance of inter-cluster load sharing is not significantly better than no load sharing.
- The performance of intra-cluster load sharing is almost as good as the performance of inter- and intra-cluster load sharing together.

As we mentioned earlier, the lack of significant improvement through inter-cluster load sharing alone, even when the overhead of job transfer between clusters is ignored, is due to the fact that the algorithm uses only the cluster macro states in making allocation decisions. This is not sufficient to yield a substantial improvement. On the other hand, the tremendous improvement achieved by employing intra-cluster load sharing is possible because each job is assigned to the least loaded node in the job's cluster. With a sufficient number of nodes per cluster, each new job can be assigned to a lightly loaded or even an idle node, which reduces the congestion in the system significantly. Performing inter-cluster load sharing, in addition to intra-cluster load sharing, results in a marginal improvement since the best node in the job's cluster is almost as good as the best node in the best cluster.

The above observations suggest that intra-cluster load sharing is far more important than inter-cluster load sharing in hierarchical systems. Since the additional benefits that can be obtained by employing the latter is small, it does not justify crossing the cluster boundaries, particularly if the transfer cost between clusters is high.

Finally, we compare the performance of intra-cluster load sharing with "join the globally least loaded node" policy in a cluster-based system. "Join the globally least loaded node" policy ignores the cluster organization of the system. It searches the

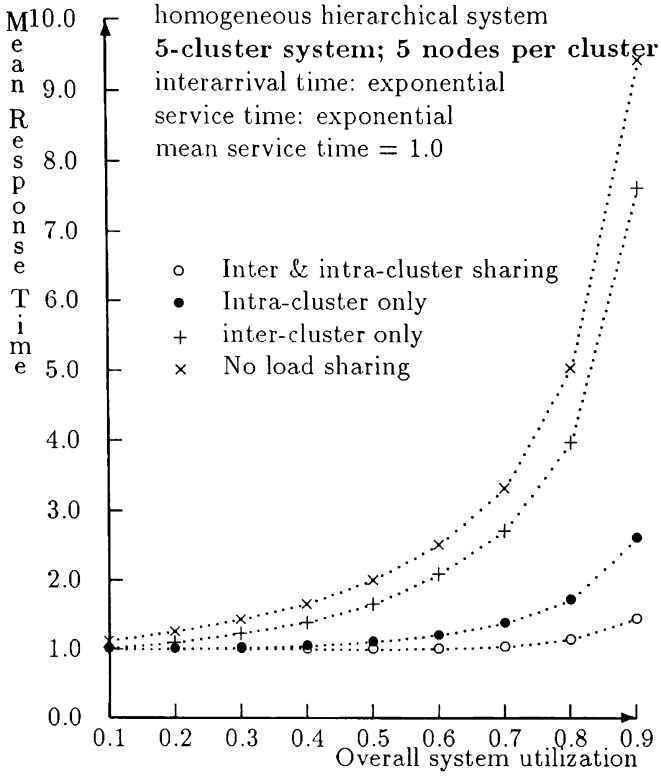

Figure 6: Load sharing policies in a cluster-based system – case (1)

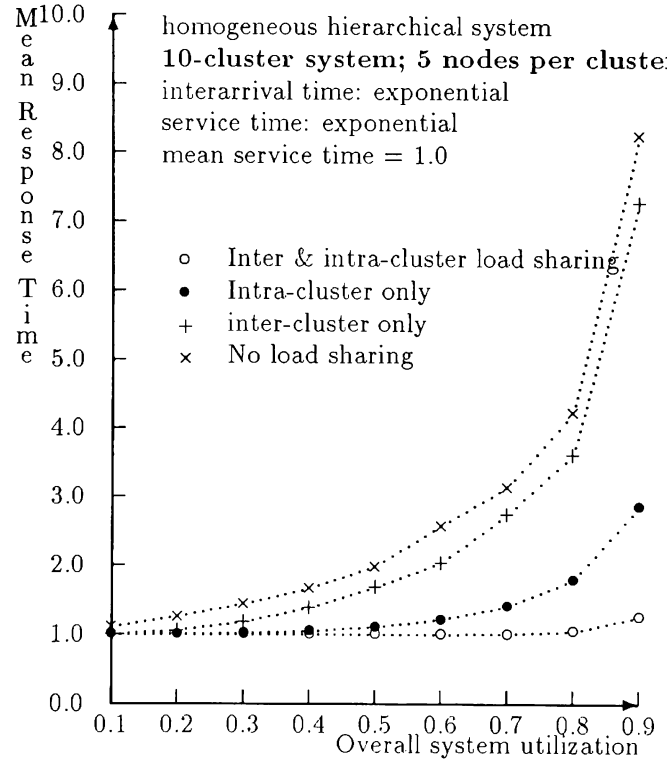

Figure 7: Load sharing policies in a cluster-based system – case (2)

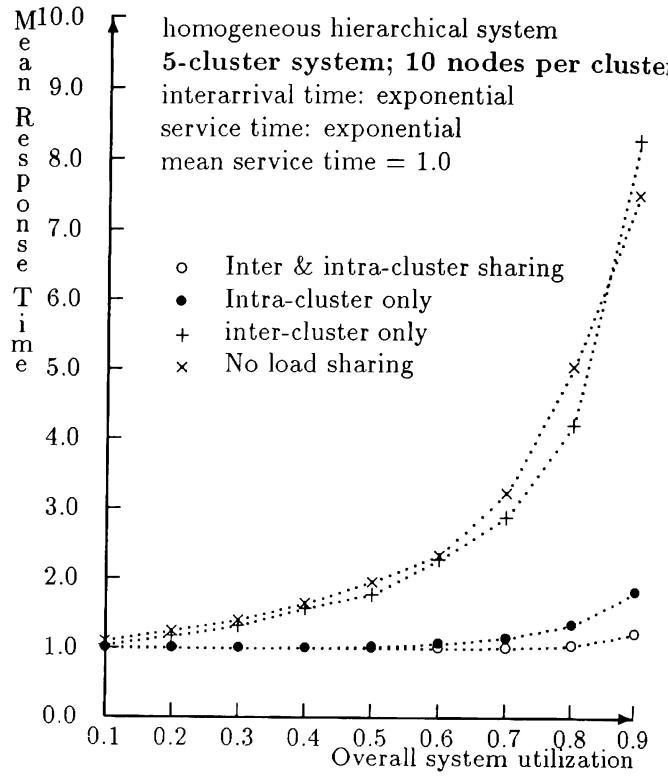

Figure 8: Load sharing policies in a cluster-based system – case (3)

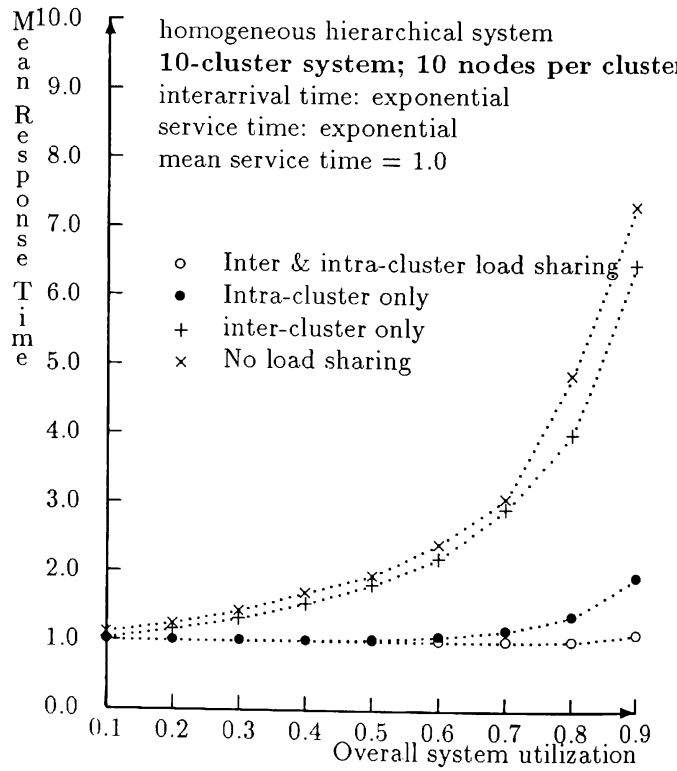

Figure 9: Load sharing policies in a cluster-based system – case (4)

whole system to find the absolutely least loaded node to which the new job is assigned. This node may belong to any cluster in the system. Note that there is a subtle difference between this policy and performing both inter- and intra-cluster load sharing. The best node in the best cluster need not be the best node in the whole system.

Since "join the globally least loaded node" ignores the system hierarchy, it is equivalent to "join shortest queue" or intra-cluster load sharing where the cluster is actually the whole system. Thus, comparing intra-cluster load sharing with "join the globally least loaded node" is, in fact, a question of the effect of the cluster size on the performance of intra-cluster load sharing. Figure 10 shows the results of simulating intra-cluster load sharing in clusters with different number of nodes. It shows that most of the benefits of load sharing can be realized with medium size clusters, ≈ 20. In this case, the contention in the system is almost eliminated. Adding more nodes can have only a minor effect on the system performance.

5 Conclusion

In this paper we considered load sharing in distributed systems with a two-level hierarchy. This organization may reflect physical proximity, administrative boundaries, etc. A two-level hierarchical system consists of clusters, each of which has its own subset of nodes. Nodes in the same cluster communicate over a local area network. Although the communication network between clusters is not specified, we assume that it is at least as costly to transfer a job between two nodes that belong to different clusters as it is if they were in the same cluster.

In a cluster-based system, load sharing may be performed within a cluster, across clusters or both. We found that load sharing across clusters alone, *inter-cluster load sharing*, does not have a significant impact on system performance since it is based on the macro state of clusters and ignores the distribution of jobs among nodes in the same cluster. This results holds even if we ignore the communication cost across clusters and regardless of the number of clusters in the system. On the other hand, allowing load sharing only within each cluster, i.e.,*intra-cluster load sharing*, can greatly improve the system performance, even if each cluster has only a moderate number of nodes. Furthermore, the performance of intra-cluster load sharing is only slightly worse than performing both *inter- and intra-cluster load sharing* together. Again, the comparison was based on the assumption that it does not cost more to transfer a job from one cluster to another than to transfer the same job between nodes in the same cluster. Should the transfer cost between clusters be higher, inter- and intra-cluster load sharing may result in system performance that is not better than intra-cluster load sharing only. We conclude that, in addition to not transferring jobs between clusters which may be costly, intra-cluster load sharing realizes most of the expected benefits of load sharing in hierarchical environments such as those found in universities and corporations.

References

[Ban87] S. A. Banawan. *An Evaluation of Load Sharing in Locally Distributed Systems.* PhD thesis, University of Washington, Seattle, August 1987.

[ELZ86] D. L. Eager, E. D. Lazowska, and J. Zahorjan. A comparison of receiver-initiated and sender-initiated adaptive load sharing. *Performance Evaluation*, 6(1):53–68, March 1986.

[ELZ88] D. L. Eager, E. D. Lazowska, and J. Zahorjan. The limited performance benefits of migrating active processes for load sharing. In *Proc. of the 1988 ACM SIGMETRICS Conf.*, pages 63–72, May 1988.

[Kor86] R. Korry. *A Load Sharing Algorithm for a Workstation Environment.* Master's thesis, University of Washington, Seattle, Washington, 1986.

[LM82] M. Livny and M. Melman. Load balancing in homogeneous broadcast distributed systems. *Performance Evaluation Review*, 11(1):47–55, 1982.

[SS84] J. A. Stankovic and I. S. Sidhu. An adaptive bidding algorithm for processess, clusters and distributed groups. In *Proc. IEEE 4th Intl. Conf. on Distributed Computing Systems*, pages 49–59, San Francisco, May 1984.

[Win77] W. Winston. Optimality of the shortest line discipline. *SIAM J. Appl. Prob.*, 14:181–189, 1977.

[WM85] Y. Wang and R. J. T. Morris. Load sharing in distributed systems. *IEEE Transactions on Computers*, C-34(3):204–217, March 1985.

[ZF87] S. Zhou and D. Ferrari. A measurement study of load balancing performance. In *Proc. IEEE 7th International Conference on Distributed Computing Systems*, pages 490–497, September 1987.

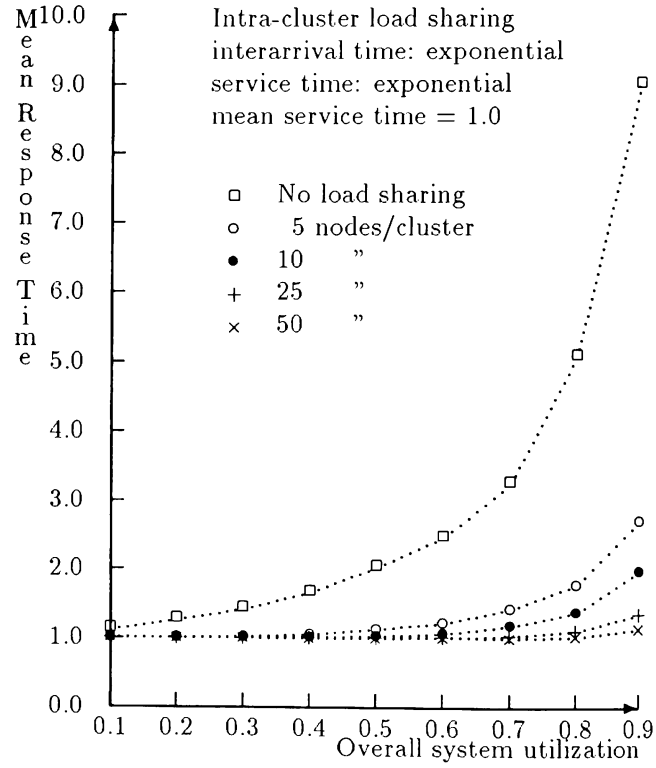

Figure 10: Intra-cluster load sharing – variable cluster size