Adaptive Resource Management

Craig E. Wills
Surendar Chandra*

Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609

Abstract: Resource information managers typically use fixed approaches to manage information. However fixed solutions to information management are complicated by the fact that computing environments are not all the same and even within a single computing environment the characteristics of the information and its access patterns may change over the lifetime of the system. These changes motivate this work to investigate an adaptive resource management mechanism that minimizes the communication overhead on a network of machines while providing resource information to clients. Our experience with a simulation study of various policies indicates a relatively simple adaptive mechanism works with further refinements needed.

1 Introduction

A class of software traditionally not viewed as configurable is that of resource information managers in a distributed environment. These servers manage information about machines, users, files and other objects. The typical approach for managing such information is to implement a fixed policy for managing the information based on historical access and update patterns along with developer experience. However fixed solutions to these problems are complicated by the fact that computing environments are not all the same and even within a single computing environment the characteristics of the information and its access patterns may change over the lifetime of the system.

For example, the introduction of an automated information access script or a performance monitor can disrupt the access patterns expected from users. This type of information access is mixed with bursty accesses from users to create a diverse set of access patterns that may change on an hourly or daily basis over the lifetime of a system. The amount and location of information may also change, which can have an impact on efficiently managing the information.

These potential changes motivate this work to investigate an adaptive resource management mechanism that minimizes the communication overhead on a network of machines while providing resource information to clients [3, 6]. The mechanism modifies its behavior according to the distribution and dynamics of the resource information, and the location and frequency of resource access. A second component of this approach is to exploit *multicasting*, which supports efficient delivery of messages to a subset of logically related machines [5]. Resources can be dynamically mapped to logical addresses for efficient delivery of resource requests or updates to only machines containing the resource or interested in the update.

The approach we take is to develop an adaptive policy based on various fixed policies and compare its performance with the fixed policies. The system incurs overhead in maintaining state information for adapting to the execution environment. However the system will provide better performance by adapting to changing resource access patterns, not only for dynamic information, but for relatively static information. Results are shown for how the policies perform in various resource access scenarios.

2 Related Work

Much work has been done in implementing systems that use fixed approaches to solve resource management problems for different problem domains. Depending on the situation, centralized servers, such as Grapevine [2], or broadcasting in a local area network, such as the Unix *rwhod* protocol, have been used as two approaches.

In terms of using multicasting, hardware support has long been supported for broadcast bus technology. Support for multicasting at the network layer for the standard Internet Protocol (IP) has been defined and implementations

*Current address is Department of Computer Science, Duke University, Durham, NC 27708.

are now available [4]. Our work assumes a local area network (LAN) environment that efficiently supports the delivery of multicast messages. In addition, systems such as ISIS [1] provide reliable multicasting to ensure that each participating distributed processes receive all messages in the same order. These approaches allow group membership to change, but support a relatively small number of groups.

Our approach is different in that not only is the group membership dynamic, but the set of multicast groups dynamically changes depending on the resources currently available in the network. One approach we have previously used with multicasting is to map resource information to multicast addresses so that a machine joins a multicast address group only if it has the corresponding resource information [7].

3 Fixed Policies

As a starting point for this work we make the observation that the solutions for information location problems in distributed system can be reduced to three classic approaches: centralize the information among one or more servers, distribute updates upon change of information, or make a query for information on a request. We use these fixed policies as a basis for our work.

Not only can broadcasting be used for distribution of updates and requests, but the latter two policies can also be used with multicasting. As we previously described, the query-on-request approach can map the resource to a logical address for delivery. We can hypothesize using a similar approach for update-on-change where updates are multicast to only those nodes interested in the changes. We can view these directed updates as replies to a *persistent queries* that are answered whenever the information changes.

Thus, with the introduction of logical addresses, these fixed policies contain some adaptive elements, but they still do not perform well if the basic access pattern assumptions are not satisfied. For example, if the number of updates is higher than the number of resource requests, the update-on-change approach sends updates which are not going to be used. These updates are wasted network packets and should be avoided.

4 Adaptive Policy

To manage a system where the patterns of information use change, an adaptive policy is explored that monitors the current access and update characteristics of the information being managed to adjust how the information is being managed. The aim is to minimize the amount of work needed to add new machines into the network while at the same time adjusting itself to the nature of the information being managed. The mechanism will adjust in three ways: how resources are mapped to logical addresses for delivery of requests and updates, whether information is distributed across all machines or centralized on a few machines, and whether information is queried on request or updated on change. How the resource information is managed is directly dependent on what type of and how frequent requests are made. Depending on the variability in how information is accessed, the system may never change, may change once and then remain stable, or it may periodically adjust how the information is managed.

The basic adaptive strategy we have explored thus far uses a distributed voting algorithm to determine if changes to the current resource management policy should be made. This approach was chosen because it allows the decision process to be distributed and each node to base decisions on easy-to-obtain parameters at the node. Each node accumulates system parameters and periodically analyzes them for the current best information management strategy. These local strategy decisions are then used as a vote for the most appropriate information management policy for the entire system. The decision votes are weighted based on the amount of local system activity from the individual nodes. For each node, if the dominating parameter is:

- *Local Accesses* or *Local Modifications*. The system makes more local accesses than remote accesses. The system should not change strategies until there appears a need for change. Since the network accesses are low anyway, the node votes to continue operating with the current scheme.

- *Modifications Sent* or *Modifications Received*. The system is handling more update packets than actually using this new information. It would be better to stop these wasteful updates and access information when there is a actual need for it. So the system votes to adapt to a query-on-request approach with requests mapped to logical addresses.

- *Requests Sent* or *Requests Received*. The system is handling requests at at a higher rate than information is modified. Thus it would be beneficial to have the information cached locally with the information originator notifying when there is a change. Thus the system votes to adapt to a update-on-change approach.

- *Replies Sent* or *Replies Received*. The system is responding to many requests implying that it would be a good node to act as a central server for this information. Thus the system votes to adapt to a centralized approach.

Once an adaptive information management strategy decision is made locally, this local decision along with the raw data that was used to arrive at this conclusion is sent to a central collecting node. This central node is not constrained to be any particular node and can be dynamically by the set of nodes.

The raw data sent by the local nodes is used by the central node to weight the votes based on the amount of machine message traffic in this individual node to arrive at a final decision. This weighting of votes is done to prevent a node with little information activity from significantly changing the global policy decision. A policy change is effected if a different, from the current, policy receives the largest number of votes and at least 50% of the total votes. The majority rule was added to prevent transient changes from affecting the system. Once a final decision is made at the central node, these decisions are sent to the individual nodes.

In our simulation study of the policy, the adaptive policy calls for a voting decision every hour. This time period is a compromise between a system that changes too fast and one that is lethargic to change. To further prevent an unstable system we explored a damping factor in the system. The damping factor is applied by comparing the current value of information accesses and modifications at each node with the same measures collected during the previous time period. If the current parameters show a change of less than 10% for both the measures this node abstains from the decision process. The individual node sends this abstain decision to the central decision node. If more than 50% of the total weight of the system abstains then the management policy will not change for lack of a majority.

5 Simulation Study

To control the experiment a simulation study was used where the system and the information access patterns are modeled. The system model we use consists of a group of workstations connected by a high speed network that supports unicast, multicast and broadcast delivery. The information model used allows instances of information to be distributed equally among all machines or to be bunched on a collection of "server" nodes. Similarly, additions, deletions and accesses of this information can be distributed evenly or unevenly among clients and servers. This information model is simple, yet loosely models realistic information and can be used to study the effect of new information access patterns. More details can be found in [6].

The various fixed and adaptive policies are studied in the context of different resource access scenarios using this information model. The system parameter of prime importance for our study is the number of "machine messages" (sent and received) as seen by the individual nodes. The length of a simulation run is for a 24 hour period. Different workloads and information access patterns are configured using an input event script that describes the distribution for an information type. The same script is used for the different fixed policies as well as the adaptive system.

The simulation runs are repeated several times and the mean values of the simulation runs are analyzed. The fixed policies exhibit little variation in behavior between runs while the adaptive policy makes discrete policy change decisions in adapting to any of the fixed resource management policies. The system is expected to adapt to the scheme which results in the fewest machine messages.

5.1 Generic Model

Before the simulation program is run through a real life workload, the system is used to simulate a simple synthetic workload. In any information access scenario we note that the amount of information modifications can be greater than, equal to, or less than the number of information accesses. The synthetic scenario successively generates accesses in all these modes. The results of this workload can be compared with the expected results for model verification.

For this scenario, we assume 20 nodes in our system and a small number of information instances at the start of the simulation run. All the nodes provide information equally and there are no server or client classes among the different nodes. The results from running this scenario for a 24 hour period are shown in Figure 1. The vertical axis indicates the total number of messages sent and received by machines over the course of the time period.

The scenario generates 10 times the information modification rate as information accesses for the first 8 hours (period I), followed by a period of equal accesses and modifications for 8 hours (period II) followed by a period of 10 times as much information accesses as information modifications (period III).

During the initial time period, when there are more information modifications as compared to information accesses, the query-on-request policy results in the smallest number of machine messages. By default, the adaptive system starts up by using the centralized information management strategy, but soon adapts to the query-on-request information access policy.

In the second phase, the adaptive policy oscillates between the query-on-request and update-on-change policies. Upon investigation, this thrashing occurs because the low system activity breaks the damping factor as the system activity is small and even a slight change in access patterns affects the decision process. This result indicates further

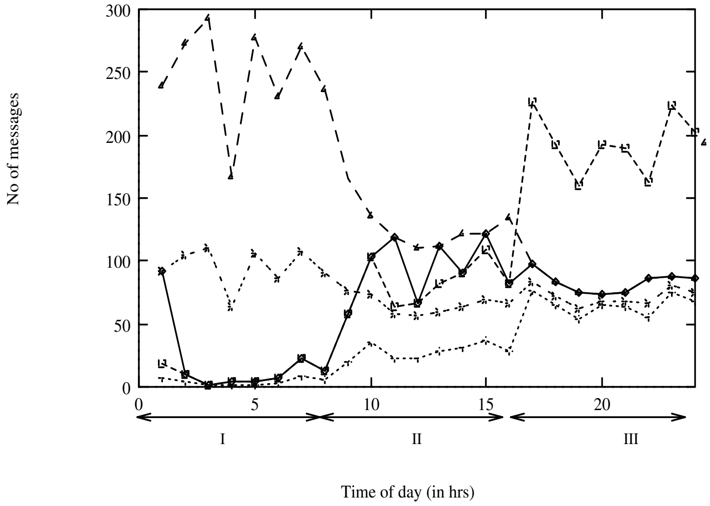

Figure 1: Machine Messages for Synthetic Information

work needs to be done on the damping factor strategy so the global effect is better controlled.

As the number of accesses increase, the update-on-change policy gives the best performance with respect to the number of machine messages. The adaptive system switches to this policy. However, because of the overhead involved in with the adaptive policy, it performs with a slightly higher cost than the fixed information management scheme.

The centralized information management policy is a good fixed policy across the different phases. However, further investigation found the adaptive policy tends to stay away from the centralized scheme because of the voting scheme. The criterion of voting for centralization is such that few of the nodes vote for it even though the policy would be best for the overall system. This aversion to centralization points to more work needing to be done on balancing individual decision making versus the global decision within the adaptive policy.

5.2 Users logged in

Overall the generic simulation model behaves as expected with the adaptive policy generally switching policies when it should. A second access scenario was created for user login information, which shows wide variance in the resource usage. The number of users logged into a system typically increases with the general working time for the day and tapers off in the wee hours. As more users log in, more requests are generated. Access patterns are affected dramatically when specific users automatically monitor other users.

For this scenario, we assume 20 nodes in our system with 100 individual users in the system. The results from running this scenario for a 24 hour period are shown in Figure 2. It shows the number of machine messages as seen by the individual nodes each hour.

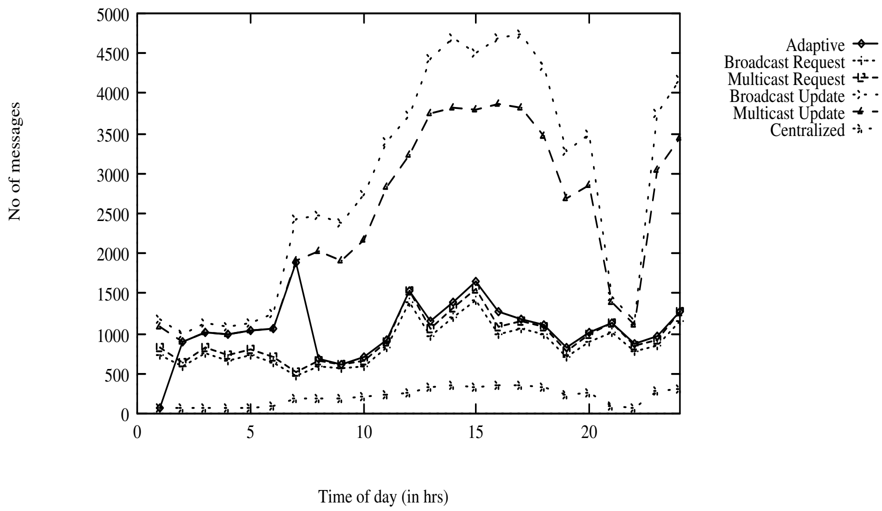

Figure 2: Machine Messages for User Information

The user activity generally increases with the time of day. Until around 6am, it remains very low and slowly increases to peak around noon. It then slowly drops off around 20:00 and drops off considerably around 22:00.

The adaptive strategy generally follows query-on-request using multicasting as the management policy. For part of the time period, the policy adapts to the update-on-change policy using multicasting until the system parameters change and the adaptive policy switches to query-on-request. Although the centralized policy shows better performance the adaptive policy again tends away from this approach. We would expect the the query-on-request with multicasting policy to yield even better results because users can be mapped to logical addresses for delivery. However the distribution of the user information does not allow the policy to take advantage of mapping users to addresses.

6 Analysis and Future Work

These preliminary results show that an adaptive policy does generally work in finding a low-cost policy in both dynamic and static environments with a relatively low overhead. More results are given in [6]. Although it does not always find the best policy, it finds a good policy to work in a range of environments. As discussed, problems did arise with the adaptive policy used with the scenarios we have generated. We do not view these problems as inherent weaknesses in the approach, but rather as directions in which further work on refining the adaptive policy needs to be done.

One direction that we need further work is how such a mechanism scales. This issue can be broken into two parts. The first is how the mechanism scales in a local area network environment with more nodes. We expect similar results, but need to test this assumption. The second issue is scaling beyond a local area network, which we believe can be done through the use of proxy requests at remote sites.

The information scenarios did bring out the problem that the adaptive policy tended away from centralization. This is a problem in the adaptive algorithm, which needs to better combine local decision making with the resulting global effect. There was also a problem in oscillations between policies when the amount of system activity is low. This decision must not only take into account the relative change of activity, but the also the amount. We also need to do more work on modeling the information. Lack of control of the information instances reduced the expected effectiveness of using logical addresses for the resources.

7 Summary

We are continuing in this effort is to explore an adaptive resource management mechanism that modifies how it works to minimize the communication overhead on network machines while providing up-to-date resource information to any client machine. The mechanism customizes itself according to the distribution and dynamics of the resource information, and the location and frequency of resource access. Whether the resource information remains distributed, or is centralized, and how a corresponding query for a resource is satisfied is the responsibility of the resource management mechanism.

The current status of our approach and preliminary results have been discussed. Our experience thus far indicates a relatively simple adaptive mechanism does work with further refinements needed. We also need to implement the policy to validate our simulation results and obtain more experience on switching information management policies.

References

[1] Kenneth Birman, André Schiper, and Pat Stephenson. Lightweight causal and atomic group multicast. *ACM Transactions on Computer Systems*, 9(3):272–314, August 1991.

[2] A. Birrell, R. Levin, R. Needham, and M. Schroeder. Grapevine: An exercise in distributed computing. *Communications of the ACM*, 25(4):260–274, April 1982.

[3] Surendar Chandra. Adaptive resource management. Master's thesis, Computer Science Department, Worcester Polytechnic Institute, August 1993.

[4] S. Deering. Host extensions for IP multicasting, August 1989. RFC 1112.

[5] Stephen E. Deering and David R. Cheriton. Multicast routing in datagram internetworks and extended LANs. *ACM Transactions on Computer Systems*, 8(2):85–110, May 1990.

[6] Craig E. Wills and Surendar Chandra. Adaptive resource management. Technical Report WPI-CS-TR-93-8, Computer Science Department, Worcester Polytechnic Institute, December 1993.

[7] Craig E. Wills and Shanti Suresh. Resource-driven resource location. In *Proceedings of the 26th Hawaii International Conference on System Sciences*, pages 80–89, January 1993.