International Journal of Soft Computing 2 (3): 422-425, 2007

Content-Based Multicasting Using JADE

B. Anandampilai
Noorul Islam College of Engineering, Kumaracoil-629180, India

Abstract: There has been a surge of interest in the delivery of personalized information to users (e.g. personalized stocks or travel information), particularly as mobile users with limited terminal device capabilities increasingly desire updated, targeted information in real time. The use of Content-Based Multicast (CBM) where extra content filtering is performed at the interior nodes of the IP multicast tree; this will reduce network bandwidth usage and delivery delay, as well as the computation required at the sources and sinks. In this study we evaluate the situations in which CBM is advantageous. This CBM is implemented using JADE. The benefits of CBM depend critically upon how well filters are placed at interior nodes of the IP multicast tree and the costs depend upon those introduced by filters themselves. Further, we consider the benefits of allowing the filters to be mobile so as to respond to user mobility or changes in user interests and the corresponding costs of filter mobility. We consider two criteria: Minimizing total network bandwidth utilization and minimizing mean information delivery delay.

Key words: Content-based, multicasting, JEDE, personalized, IP, utilization

INTRODUCTION

The basic multicast does not concern itself with the content or structure of the information being delivered. The limitations of basic IP multicast become much more severe once the recipients desire more complex filtering and personalization and especially if the information being delivered is unstructured or has limited meta-data. For instance, one recipient of an Internet radio channel may desire to block out all advertisements, except those for cars, which she is currently in the market for; another likes country music but cannot abide certain artists or songs; a third wants news but no more about the latest celebrity scandal; and so on. Even if the information is well-structured,e.g. consider a NYSE stock quotes example, users can desire complex filtering e.g. Inform me only if IBM stock changes by more than 10% in price or 5% in volume compared to yesterday. We propose the paradigm of Content-Based Multicast (CBM) (Lee *et al*., 2000). CBM takes into account the structure and semantics of the information being disseminated and attempts to minimize both network utilization as well as the processing at the recipients by intelligently filtering the information as it propagates towards each recipient. We consider a solution where the filtering is done by means of filters, which can potentially be mobile, residing at (a subset of) the intermediate nodes in the network, either in IP or application-level routers or at attached CBM servers. A filter can then apply complex criteria and ensure that information propagates down the tree to a child only if a user at a leaf in that child's sub tree desires the information. As users move or change their filtering criteria, or as the information being disseminated changes, filters may move from one interior node to another in response. While filters thus increase multicast efficiency they also introduce additional computation, complexity and delay in the multicast process (Ko *et al*., 2003) and hence the number of filters in the network should be limited.

Example: We use the IP multicast tree to illustrate the benefits of CBM. The subscription requests of each recipient (leaf), labeled A-I, for items (numbered 1 to 8 in this example) are shown along the bottom of the tree. These subscriptions propagate up the tree; interior n odes are labeled with the union of the subscriptions from the leaves in their sub tree. The source sends the 8 (equal-sized) items of information requested by at least one recipient

SYSTEM MODEL

We assume for simplicity that there is a single source of information connected by an IP network to a set of recipients. The source periodically receives updated information to be disseminated. We assume that an IP multicast tree has been set up using an appropriate protocol; the root is the source and the leaves are the recipients. For CBM we assume that there is a set of ssoftware modules, called filters, distributed at the interior nodes of this multicast tree. The filters reside at filter platforms, which can be IP routers or at servers attached to the local subnet of a router. Here we can utilize Mobile Agents as filters. The idea of employing Mobile Agents for applications such as distributed information retrieval, performance monitoring and remote data filtering or aggregation has been extensively reported in the literature A Mobile Agents is essentially an autonomous object containing the logic to perform a given task and possibly migrate under its own control from node to node in a network.

EFFECTIVE LOCATION OF FILTERS

The JADE Agent location problem consists of two phases. Initially, we need to determine the appropriate number of agents for a given monitoring problem and compute the location of each of those agents. Subsequently, upon JADE Agent deployment the JADE Agent system needs to be able to self-regulate in order to adapt to changing conditions. This is achieved by triggering JADE Agent migration in a controlled fashion to avoid instability due to continuous JADE Agent (www.cs.cm.edu) migration. The problem of computing the optimal number and location of area monitors is analogous to the optimal placement of *p* servers in a large network, In the system proposed herein, the location of area monitors is neither fixed nor predetermined at design time. Area monitors are realized with MAs, simple autonomous software entities that, having access to network routing information, can adapt and roam through the network. The distributed monitoring system is deployed by progressively partitioning the network and populating each partition with monitoring agents. We assume the existence of an JADE Agent system supporting mobility and cloning. Agents are assumed to have access to routing information obtainable from network routers through standard network management interfaces. For simplicity, we also assume that MA hosts (i.e., locations in which Mas are able to run) are evenly distributed within the network. This, in other terms, means that for each router there is always an MA host that is located relatively close to it and, for each LAN, the number of MA hosts is proportional to the number of Monitored Objects (MOs) that need to be monitored in that LAN. Under these assumptions, the MA distribution tree (i.e., the set of routes used for MA deployment) does not differ significantly from the routing tree rooted at the monitoring station. Without loss of generality, we envisage a scenario in which routers can act as MA hosts during MA deployment. In such case, the MA distribution tree would a ctually coincide with the routing tree.

JADE AGENT DEPLOYMENT

The basic underlying idea used to solve the JADE Agent location problem is to exploit information that is readily available in the system rather than trying to derive a new network distance matrix. This is why our monitoring system relies solely on routing information, which is maintained by routing protocols. The precise nature and quality of this information will depend on the routing protocol in use. Sophisticated routing protocols such as Open Shortest Path First (OSPF) can maintain information using multiple distance metrics (delay, throughput, etc.). Our objective is to design a deployment algorithm that will optimize JADE Agentlocation with respect to whatever metric information is available. The algorithm deploys the area monitors (or MAs) during the network partitioning process through a clone and send process starting at the main monitoringstation. The number and location of MAs is computed by subsequently comparing the monitoring task parameters with routing information extracted from network routers. For the sake of simplicity, we assume that the list of MOs consists of all the network nodes. The algorithm deploys the area monitors (or MAs) during the network partitioning process through a clone and send process starting at the main monitoring station. The number and location of MAs is computed by subsequently comparing the monitoring task parameters with routing information extracted from network routers. The monitoring task, including the list of MOs as well as the operations to be performed on them, is delegated to a first MA.

JADE AGENT DEPLOYMENT ALGORITHM

This MA (www.multiagent.com) starts executing the algorithm, which is distributed in nature; that is, each subsequently cloned MA will execute the same algorithm (only the input configuration parameters will differ). Starting from the monitoring station, the initial MA builds an estimate of the total monitoring cost on the basis of the individual cost or distance field of the local routing table (i.e., for each element of the MO list, its routing cost will be added to the total estimated monitoring cost). Having estimated the total cost for performing the task from its current location, the MA will then consider alternative configurations using partial costs attached to the neighbor nodes. Not all neighbor nodes need to be discovered: Only those nodes that are next hops for the MOs targeted by the MA are candidate MA hosts. They will be discovered, again, through the local routing table; that is, they are identified by next-hop addresses of those entries whose destination field contains one of the targeted MOs. The partial cost attributed to a neighbor node is found by adding the individual costs (extracted from the table) of those MOs that are targeted by the MA and reached through the node. At this point, a simple heuristic function is used to decide the number of new agents to be cloned and their initial location (among the neighbor nodes). This function estimates the need to send or clone an MA to a neighbor (i) on the bases of the total number of MOs targeted by the JADE Agent($|MO|$), the number of Mos reached through the node ($|MOi|$), the total Cost (Ct) and the partial Cost (Ci). The cloning Threshold value (Th)impacts the final number of Mas.The application of JADE Agents in our system is due to its characteristics and lets see some of its characteristics.

JADE-AN OVERVIEW

JADE (Genesereth and Ketchpel, 1994) (Java Agent Development Framework) is a software development framework aimed at developing multi-agent systems and applications conforming to FIPA (Foundation for Intelligent Physical Agents) standards for intelligent agents. JADE is a middleware that facilitates the development of multi-agent systems. It includes:

- A runtime environment where JADE agents can live and that must be active on a given host before one or more agents can be executed on that host.
- A library of classes that programmers have to/can use (directly or by specializing them) to develop their agents.
- A suite of graphical tools that allows administrating and monitoring the activity of running agents.

Each running instance of the JADE runtime environment is called a Container as it can contain several agents. The set of active containers is called a Platform. A single special Main container must always be active in a platform and all other containers register with it as soon as they start. It follows that the first container to start in a platform must be a main container while all other containers must be "normal" (i.e., non-main) containers

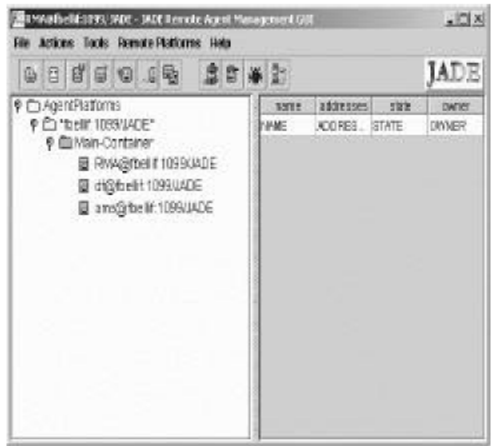

Fig. 1: Jade GUI

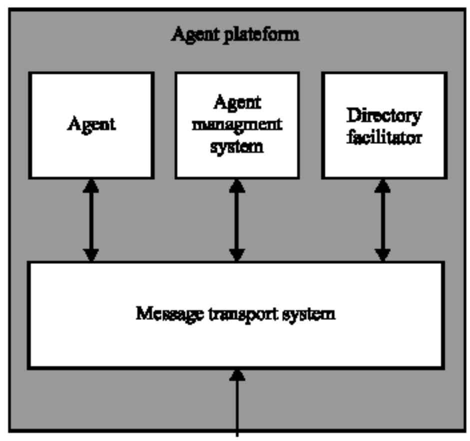

Fig. 2: Reference architecture of a FIPA agent platform

and must "be told" where to find (host and port) their main container (i.e. the main container to register with). The agent platform provides a Graphical User Interface(GUI) for the remote management, monitoring and controlling of the status of agents, allowing, for example, to stop and restart agents. The GUI allows also to create and start the execution of an agent on a remote host, provided that an agent container is already running. The GUI allows also to control other remote FIPA-compliant agent platforms (Fig. 1).

The agent platform can be distributed on several hosts. Only one Java application and therefore only one Java Virtual Machine (JVM), is executed on each host. Each JVM is basically a container of agents that provides a complete run time environment for agent execution and allows several agents to concurrently execute on the same host. JADE agents are identified by a unique name and provided they know each other's name, they can communicate transparently regardless of their actual location.The standard model of an agent platform, as defined by FIPA, is represented in Fig. 2.

CONCLUSION

The use of Content-Based Multicast (CBM) where extra content filtering is performed at the interior nodes of the IP multicast tree will reduce network bandwidth usage and delivery delay, as well as the computation required at the sources and sinks. Thus we evaluated the situations in which CBM is advantageous. The implementation of CBM using JADE Agents makes it more efficient.

REFERENCES

Genesereth, M.R. and S.P. Ketchpel, 1994. Software agents. Commun. ACM., 37: 48-53.

Ko, Y.B., S.J. Lee and K.Y. Lee, 2003. A multicast protocol for physically hierarchical Ad Hoc networks, Proc. IEEE VTC, Jeju, Korea.

Lee, S.J., W. Su, J. Hsu, M. Gerla and R. Bagrodia, 2000. A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols, Proc. IEEE Infocom'00, Tel-Aviv, Israel.

www.multiagent.com

www.cs.cmu.edu