IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661 Volume 3, Issue 6 (Sep-Oct. 2012), PP 36-38
www.iosrjournals.org

A middleware approach for high level overlay network

Mary Metilda. J

Assistant Professor, Department of CSE, Roever Engineering College, Perambalur - 621 212

Abstract: *In network management middle wares are required to enhance reliability and efficiency of distributed systems. The proposed approach uses swarm-intelligence networks at the monitoring level to provide self-organization, adaptability, scalability and reactivity features. Resource management is performed by a multi-agent based layer. Knowledge exchange between the monitoring and management layers is mediated by a data warehouse.*

General Terms: *Management, Performance, Reliability, Algorithms.*

Keywords: Intelligent middleware, Network Management, Swarm intelligence, Ant algorithms.

I. Introduction

The design of distributed applications for large scale dynamic networks, such as ad-hoc networks or wireless mesh networks, introduces several challenges which cannot be entirely solved by traditional network management techniques. Low-level network management is not sufficient to ensure reliable and persistent connectivity of pervasive devices, making it difficult to meet user specified Quality of Service (QoS). It is possible to further improve the performance of existing mechanisms for known low-level problems like network devices supervision, higher level solutions may prove to be more effective and simpler to implement. High-level management is typically tied to specific information and requirements, and is likely expected to deal with more abstract problems. It is thus worth implementing intelligent high-level network management middleware's that the dynamic, heterogeneous and volatile nature of the network to provide a robust, reliable, and efficient platform to meet requirement. Those goals are imply proactive and reactive behaviors, prediction and the capability environmental changes. This paper presents a modular approach to high-level network management, which aims at supporting various kinds of complex distributed systems and overlay networks, such as computing grids or Peer to Peer systems. Whereas the traditional network management relies on complex, task specific appliances and protocols, the proposed architecture completely separates the network monitoring part from the management logics.

The multilayer architecture is composed of a higher resource management layer and a lower ant-based monitoring layer. The upper layer uses information gathered by the lower one in order to accomplish their tasks of controlling, coordinating, and allocating network resources. An additional intermediate layer provides an interface for information exchange. The monitoring activity is based on the ant colony paradigm, which has already proved to be a suitable solution to a wide range of network management related problems, such as routing, resource discovery, and load-balancing. Ant colony optimization is a branch of swarm intelligence inspired by the social behavior of ants. Artificial ants are simple mobile software agents that move across the network and collect information about visited nodes, such as available resources or system status. The adoption of a swarm intelligence approach in the signaling layer aims at providing the required self organization, adaptability, scalability, and reactivity features to the platform.

The resource management layer is implemented upon distributed agent based architecture. Intelligent agents provide an interface to the user, and fulfill the requests by the mean of coordination and collaboration mechanisms. The ability of each agent to adapt and learn increases the efficiency and robustness of the system. Both layers are independent from each other, and it can be easily replaced or adapted to different management requirements and scenarios, ranging from wireless networks to computing grids or sensor networks. To enforce this independence, interaction between the signaling and management layers is mediated by a data warehouse layer that provides stateful information.

II. System Analysis

The present's system work in the domain of adaptive and self-organizing network management middleware's. Although research on adaptive middleware's provides several examples of self-organizing solutions for both low and high-level network management, like in, the focus here is on self organization through the social insect paradigm. An example of low-level management inspired by a biological behavior can be found in the Ecomobile middleware. Ecomobile implements a bio-inspired method to configure, monitor, analyze, and control optical transport networks. The system architecture is based on different kinds of mobile agents that separate the navigation and coordination tasks from the computational one. Monitoring and management activities are performed at the same level: information is collected, moved, exchanged, and exploited directly by agents through a coordinated behavior. Concerning high-level management, Messor and ARMS. Both architectures are completely uncoupled and highly modular. Messor is aimed at P2P systems, and achieves self-organization by using an ant-based algorithm to perform load balancing between nodes. Ants are mobile agents that can move across the network, access services made available by visited nodes, and monitor their workload. Despite the simple logics behind single ants, the emergent behavior of the whole colony allows quick balancing of the system.

An ARM is a multi-agent grid computing middleware that supports process scheduling and load balancing. Agents are organized hierarchically and can exchange information about available resources and forward incoming requests that they cannot satisfy. In order to find the best match for every request, resources and applications are modeled by the middleware. An ARM has been subsequently improved to use an ant algorithm similar to Messor for the load balancing task. In ARMS, ants are just XML documents that carry information about node workload, whereas agents act as nests and implement the logics of the ant algorithm

III. SmartG System Architecture

SmartG is a generic and modular middleware that aims at supporting intelligent network management using swarm intelligence algorithms for the monitor task. With respect to existing projects presented in the previous section, which were all application-specific, SmartG is flexible and easily adaptable to different scenarios. The monitoring and management activities can be customized independently depending on both the requirements of the application, and the nature of the underlying network.

The middleware architecture is composed of two main layers connected by the mean of an intermediate data warehouse layer. Figure 1 shows an overview of this design: the Smart Signaling Layer (SSL) is located at bottom; the Smart Resource Management Layer (SRML) is on top. Interaction between the top and bottom layers is mediated by a data warehouse; this architecture may be considered as loosely coupled, because components are able to operate independently in the event of a communication failure. This section describes each layer and its functionality.

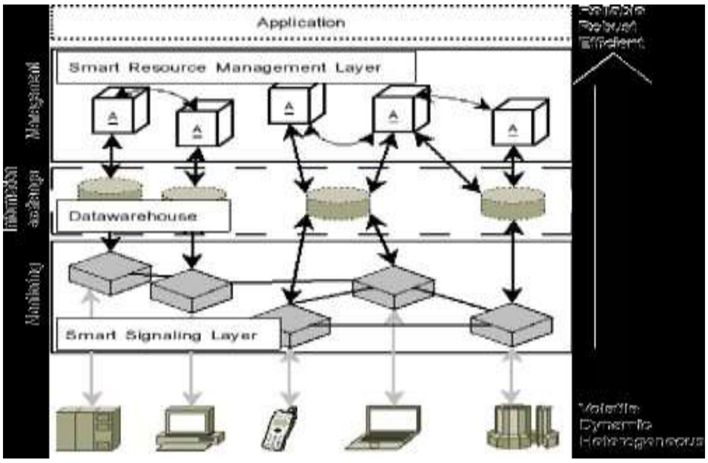

Figure 1: SmartG architecture

3.1 Smart Signaling Layer

The SSL is an incharge of retrieving information about the network status and storing them in the data warehouse. This layer operates in a completely decentralized way, achieving its monitoring task by mean of mobile swarm agents. Each node of the network acts as a nest by running a software application that provides an environment for the execution of ant-colony algorithms. The advantage of such a distributed approach is an increase in the robustness and adaptability of the system. Ants moving across the network retrieve information about visited resources such as their status, availability or location. Collected information is stored locally in the node, to be used by other individuals in the colony, or directly in the data warehouse. Birth and death of ants are triggered by events coming from devices, requests made by other individuals, and by changes in data stored in the data warehouse. At this level, network topology is highly dynamic, and requires a fully connectionless design between nodes.

The concept of node neighborhood depends on the underlying system: in P2P systems the actual topology has a one-to-one correspondence with the overlay network, whereas in mobile ad-hoc networks it may depend on the physical position and proximity of devices. Several different types of colonies can be executed on the platform to gather and spread information required by resource management processes such as discovery, routing and load balancing. The signaling layer implementation has to be lightweight enough in order not to interfere with normal network operations. A lightweight design also has the advantage of being able to run on small devices with limited capabilities, in order to support heterogeneous networks consisting of both fixed and mobile devices (for example, notebooks, PDA, mobile phones).

3.2 Smart Resource Management Layer

The SRML manages higher-level tasks of the middleware typically using application-specific intelligent agents that can collaborate and coordinate to meet application requirements. Agents work both upon receiving new requests from the application, and in an autonomous way, to continuously assure QoS of running processes; if a request cannot be fulfilled, an agent can forward it to its neighbors. The decision process is driven both by local information saved by the agent itself, and some global information retrieved by the monitoring layer and stored in the data warehouse.

The type of required information heavily depends on the goal to achieve: as an example, service discovery may require a model of the requested service, a list of available offers matching the request, and the cost of reaching them. As the systems aims at being adaptive in respect of the high level requirements and low-level resource status, full interaction between the SRML and applications is required. On the one hand, applications must be able to communicate with the management layer to request services and resources, specify QoS requirements, and control the middleware.

On the other hand, management agents should be able to interact with the application not only to provide the requested data, but also to rearrange and adapt requirements with respect to actual resource availability. For example, a resource discovery agent can store the parameters of a request and inform the corresponding application as soon as resources that better fulfills quality levels are available.

3.3 Data warehouse

The data warehouse layer mediates communication between the SSL and SRML and provides stateful information storage and knowledge exchange to both layers. Different approaches to implement a distributed data warehouse can be considered depending on the usage scenario. In highly dynamic networks the flow of information coming from the underlying layer can be important and could represent an issue. In these situations, high performance databases are required to avoid bottlenecks and delays, and ensure reliability. On the other hand, mobile platforms require local smaller databases or local caching, to optimize communication between the device and the network.

IV. Conclusion

Actual implementation of the model is conducted in the framework of a grid management middleware, to evaluate the viability of the approach in large scale environments. The proposed model aims at increasing the reliability and efficiency of distributed systems by mean of proactive and reactive management. The multi-layer architecture completely separates the monitoring from the management tasks, allowing customizing the middleware to different scenarios. The lightweight network activity uses ant-colony algorithms and is able to self-organize and rapidly react to environmental changes. Resource management is performed by intelligent agents using information gathered by the monitoring layer. A data warehouse layer is used to interface monitoring and management and to provide a knowledge exchange channel.

References

[1] Akyildiz, I. F., Wang, X., *A Survey on Wireless Mesh Network,* Sept. 2005.

[2] Janacik, P., Kao, O., Rerrer, U., *A routing approach using swarm-intelligence for resource sharing in wireless and ad hoc networks*, 2004.

[3] Rossier-Ramuz, D., *Towards Active Network Management with Ecomobile, an Ecosystem-inspired Mobile Agent Middleware*. October 2002.

[4] Hirsbrunner, B., Courant, M., Brocco, A., Kuonen, P., *SmartGRID: Swarm Agent-Based Dynamic Scheduling for Robust, Reliable, and Reactive Grid Computing*, October 2006.

[5] Brocco, A., Hirsbrunner, B., Courant, M., *Solenopsis: A Framework for the Development of Ant Algorithms*, 2007.

[6] diuf.unifr.ch/pai/solenopsis.

[7] www.haslerstiftung.ch.