A Modular Middleware for High-level Dynamic Network Management

Amos Brocco
PAI Group
Department of Informatics
University of Fribourg
+41.26.300.8481
amos.brocco@unifr.ch

Béat Hirsbrunner
PAI Group
Department of Informatics
University of Fribourg
+41.26.300.8467
beat.hirsbrunner@unifr.ch

Michèle Courant
PAI Group
Department of Informatics
University of Fribourg
+41.26.300.8470
michele.courant@unifr.ch

ABSTRACT
In addition to the control and supervision of components and connections, network management middlewares are required to enhance reliability and efficiency of distributed systems. In this paper, we describe a modular and distributed middleware architecture aimed at high-level dynamic network management. The proposed approach uses swarm-intelligence 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 datawarehouse. The model is evaluated through a case study for grid computing.

Categories and Subject Descriptors
D.2.11 **[Software Architectures]**: Domain-specific architectures.

I.2.11 **[Distributed Artificial Intelligence]**: Intelligent agents, Multiagent systems.

C.2.4 **[Distributed Systems]**: Distributed applications

General Terms
Management, Performance, Reliability, Algorithms.

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

1. INTRODUCTION
The design of distributed applications for large scale dynamic networks, such as ad-hoc networks or wireless mesh networks [1], 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). Notwithstanding that it is certainly 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 application specific information and requirements, and is likely expected to deal with more abstract problems. It is thus worth implementing intelligent high-level network management middlewares that abstract the dynamic, heterogeneous and volatile nature of the network to provide a robust, reliable, and efficient platform to meet application requirement. Those goals imply proactive and reactive behaviors, prediction and avoidance of critical circumstances, the capability to rapidly adapt to environmental changes, and the ability to cope with unexpected situations.

This paper presents a modular approach to high-level network management, which aims at supporting various kind of complex distributed systems and overlay networks, such as computing grids or P2P systems. Whereas traditional network management relies on complex, task specific appliances and protocols (such as SNMP [2]), the proposed architecture completely separates the network monitoring part from the management logics. The multi-layer 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 its 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 [3], which has already proved to be a suitable solution to a wide range of network management related problems, such as routing [4], resource discovery [5], and load-balancing [6][7]. Ant colony optimization is a branch of swarm intelligence [8] 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. Ants can communicate indirectly with each other by leaving pheromone trails, replicating a behavior exhibited by real ants called stigmergy. 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 a distributed agent based architecture. Intelligent agents provide an interface to the user, and fulfill 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 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 datawarehouse layer that provides a stateful information storage.

The remainder of this paper is organized as follows: Section 2 summarizes some of the related work in the domain of network management and middleware applications; Section 3 presents the SmartG middleware architecture, whereas Section 4 discusses an actual instantiation of the framework for grid computing. Finally, Section 5 presents some conclusions and directions for further work.

2. RELATED WORK

This section presents some related work in the domain of adaptive and self-organizing network management middlewares. Although research on adaptive middlewares provides several examples of self-organizing solutions for both low and high-level network management, like in [9][10], 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 [11] middleware. Ecomobile implements a bio-inspired method to configure, monitor, analyze, and control optical transport networks. The 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, two projects can be considered as the most related to the work presented in this paper: Messor [6] and ARMS [12]. Both architectures are completely uncoupled and highly modular.

Messor is a grid computing system for highly-parallel and time-intensive computations built upon the Anthill [13] framework. 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. When an overloaded node is found the ant tries to find an underloaded one: on success, it requests the former to off-load jobs from the latter. Despite the simple logics behind single ants, the emergent behavior of the whole colony allows quick balancing of the system.

ARMS 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. ARMS has been subsequently improved [14] 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.

3. SMARTG 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 datawarehouse 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 datawarehouse; 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.

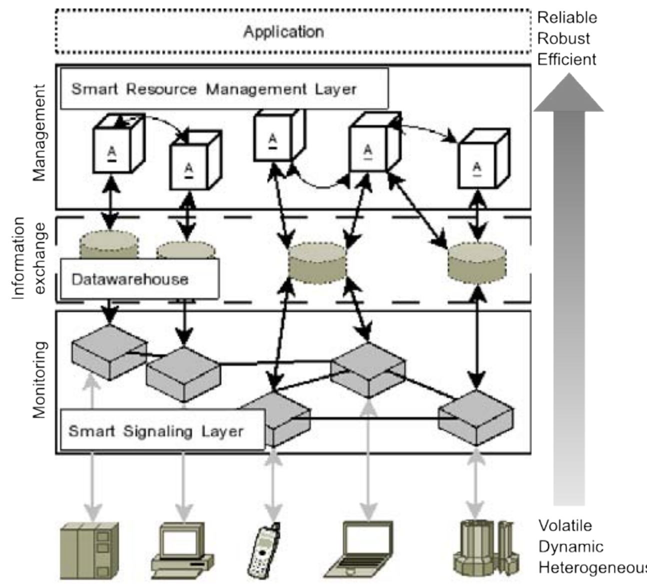

Figure 1: SmartG architecture

Smart Signaling Layer

The SSL is in charge of retrieving information about the network status and storing them in the datawarehouse. 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 are stored locally in the node, to be used by other individuals in the colony, or directly in the datawarehouse. 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 datawarehouse.

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).

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 datawarehouse. 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 both the high-level requirements and low-level resource status, full interaction between the SRML and applications is required. On 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.

Datawarehouse

The datawarehouse 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 datawarehouse 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.

4. CASE STUDY

For validation purposes, the proposed modular middleware architecture, has been instantiated in the SmartGRID grid management middleware [15]. The following subsections provide the motivation behind this specific middleware, an insight of actual development status, an evaluation of current achievements, as well as future works.

Background

Grid computing introduces several challenges that are nowhere to be found in conventional distributed computing infrastructures; large scale networks, heterogeneous devices, and extremely dynamic behaviors make it inherently difficult to allocate resources and exploit the available computing power.

Given that centralized management is difficult to maintain and limits the scalability of such systems, the goal of a Grid middleware is to provide a distributed and modular solution to guarantee reliability, efficiency and robustness. Following the proposed architecture, the objective is to provide a way to achieve optimal resource management by monitoring the network and computing infrastructure in a reactive way and by applying an intelligent global scheduling policy.

Design

The SSL layer is implemented by the Solenopsis framework [16][17]. Solenopsis is a Java platform that allows to run different types of algorithms, to collect information required by resource discovery, scheduling and load balancing activities. The logics of each algorithm is carried by the ant itself, thus easing deployment of different species with different tasks.

Current development focuses on load balancing algorithms: ants are created on overloaded nodes and wander around searching for free nodes. As soon as a suitable candidate is found, the information is passed to the management layer via the datawarehouse, in order to complete the load balancing. Other SSL activities such as resource discovery, and path discovery will be investigated in the future.

The datawarehouse is based on multiple relational databases containing information about the computing resources, such as power, workload, and status of running processes.

The SRML is in charge of process scheduling and resource allocation, including task migration, and local priority management. Scheduling decisions must favor a global optimal usage of the grid on the basis of partial knowledge gathered from lower layers.

Evaluation

The resource management layer ensures a stable behavior and prevents management-monitoring coupling situations, which can occur if the upper layer overreacts, or reacts too rapidly to events and changes coming from the lower layer.

Due to the heterogeneous and dynamic nature of the considered scenario, some reliability and security concerns remain to be addressed.

In order to gather as much information as possible, the signaling layer spreads across the whole network, possibly overcoming private network domains. As agents can move during execution, the platform must ensure that visiting agents cannot perform malign actions on a node. Current implementation guarantees security by running ants in sandboxed environments, and by giving them access to only a limited and controlled set of resources. Other foreseeable solutions would require adding trust mechanisms to certify the origin of an agent and ensure its dependability, or the introduction of immune systems techniques.

Further work would focus on efficiency of the scheduling algorithm and of network usage, to quantify the system overload caused by the monitoring activity and by process migration.

5. CONCLUSION

This article presents a modular and adaptive middleware solution for high-level network management. The proposed model aims at increasing the reliability and efficiency of distributed systems by mean of proactive and reactive management.

Given that centralized solutions are not suited for large scale networks and highly dynamic environments, a completely distributed and modular approach has been taken.

The multi-layer architecture completely separates the monitoring from the management tasks, allowing to customize the middleware to different scenarios. The lightweight monitoring 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 datawarehouse layer is used to interface monitoring and management and to provide a knowledge exchange channel.

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.

6. ACKNOWLEDGMENTS

This project is developed thanks to the financial support of the Swiss Hasler Foundation [18] in the framework of the "ManCom Initiative" (ManCom for "Managing Complexity of Information and Communication Systems"), project Nr. 2122.

7. REFERENCES

[1] Akyildiz, I. F., Wang, X., *A Survey on Wireless Mesh Networks*. IEEE Communications Magazine, vol. 43, no. 9, s23-s30, Sept. 2005.

[2] RFC 1157, A Simple Network Management Protocol (SNMP), http://tools.ietf.org/html/rfc1157

[3] Dorigo, M., Stützle, T., *Ant Colony Optimization.* MIT Press, 2004.

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

[5] Wittner, O., Heegaard, P.E., Helvik, B.E., *Scalable Distributed Discovery of Resource Paths in Telecommunication Networks using Cooperative Ant-like Agents.* Congress on Evolutionary Computation (CEC2003), December 8-12, Canberra, Australia.

[6] Montresor, A., Meling, H., *Messor: Load-Balancing through a Swarm of Autonomous Agents.* Technical Report UBLCS-02-08, Dept. of Computer Science, University of Bologna, May 2002.

[7] Salehi, M.A., Deldar, H., *Grid Load Balancing using an Echo System of Intelligent Ants.* Proceeding (517) Parallel and Distributed Computing and Networks, 2006.

[8] Bonabeau, E., Dorigo, M., Theraulaz, G. , *Swarm Intelligence : From Natural to Artificial Systems (Santa Fe Institute Studies on the Sciences of Complexity).* Oxford University Press, USA, 1999

[9] Kurzyniec, D., Wrzosek, T. Drzewiecki, D., Sunderam, V., *Towards Self-organizing Distributed Computing Frameworks: The H2O Approach.* Parallel Processing Letters, Vol. 13, No. 2, pp. 273-290, 2003.

[10] Liabotis, I., Prnjat, O., Olukemi, T., Li Mow Ching, A., Lazarevic, A., Sacks, L., Fisher, M., McKee, P., *Self-organising management of Grid environments.* International Symposium on Telecommunications, Isfahan, Iran, 2003

[11] Rossier-Ramuz, D., *Towards Active Network Management with Ecomobile, an Ecosystem-inspired Mobile Agent Middleware.* Department of Informatics, University of Fribourg, PhD Thesis n° 1392, October 2002.

[12] Cao, J., Jarvis, S.A., Saini, S., Kerbyson, D.J., Nudd, G.R. *ARMS: An agent-based resource management system for grid computing.* Scientific Programming 10 (2002) 135–148

[13] Babaoğlu, Ö., Meling , H., Montresor, A., *Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems*. In Proc. Of the 22^{th} Int. Conf. On Distributed Computing Systems, Vienna, Austria, July 2002.

[14] Cao, J., *Self-Organizing Agents for Grid Load Balancing.* Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing (GRID '04), 2004.

[15] Hirsbrunner, B., Courant, M., Brocco, A., Kuonen, P., *SmartGRID: Swarm Agent-Based Dynamic Scheduling for Robust, Reliable, and Reactive Grid Computing*. Working Paper 06-13, Department of Informatics, University of Fribourg, Switzerland, October 2006.

[16] Brocco, A., Hirsbrunner, B., Courant, M., *Solenopsis: A Framework for the Development of Ant Algorithms.* IEEE Swarm Intelligence Symposium, Honolulu, USA, 2007. *Camera Ready*

[17] Solenopsis Framework Website, diuf.unifr.ch/pai/solenopsis

[18] Swiss Hasler Foundation Website, www.haslerstiftung.ch