A Middleware for Large Scale Networks Inspired by the Immune System

M. Bakhouya, J. Gaber and A. Koukam
Laboratoire SeT
Université de Technologie de Belfort-Montbéliard
90000 Belfort, France
{bakhouya, gaber, abder.koukam}@utbm.fr

Abstract

Very large scale networks such as the Internet require a new operational model to use resources efficiently and reduce the need for the administration necessary in client-server networks. In this paper, we present an autonomous decentralized system inspired by the immune system as an alternative to the traditional client-server paradigm. The immune system has a useful set of organizing principles that guide the design of scale, adapt and efficient enough networking model to bring answers to some large scale networking challenges.

1. Introduction

Very large scale networks such as Internet are heterogeneous, continuously increasing in size and complexity, making their management difficult to achieve. The number of resources will be nomadic mobile and partially connected. The rule today is large scale networking where billions of users access variety and increasing number of resources distributed over the network. The ability to maintain and allocate all of these resources is increasingly difficult. The traditional client-server approaches become impracticable to surmount this difficulties. Many effort are now developed to propose decentralized approaches for networks management, and many of them are based on mobile agents[2, 4, 10, 16]. Issues are that network services must be able to scale to billions of nodes and users, able to adapt to dynamic conditions in the network, must be highly available, secure and should require minimal Human configuration and management[5, 18].

In this paper, we present an autonomous decentralized system based on mobile agents and inspired by the immune system as an alternative to the traditional client-server paradigm. The immune system has a set of organizing principles such as scalability, adaptability and availability that are useful for developing a networking model in highly dynamic and instable setting.

A fundamental aspect of our model is the process of resource allocation. In the traditional Client/Server approach, to use a resource, the user initiates a call by submitting a request to a server that manages and controls this resource. Unlike the classical C/S approach, in our immune-based approach, each user request is considered as an attack launched against the global network. The immune networking middleware reacts like an immune system against pathogens that have entered the body. **It detects the infection (i.e., user request) and delivers a response to eliminate it (i.e., satisfy the user request)**. This immune approach can be considered as the opposed approach to the client-server one.

In our approach, immunity-based agents roam around the machines and routers, and allocate and/or recognize resources distributed across the network. These agents can mutually recognize each client requests and can take appropriate actions to localize the required resources. Such agents can learn and adapt to their environment dynamically and detect both known(self) and unknown (non-self) resources. This research is the part of an effort to develop a mobile agent-based middleware that can monitor the resources distributed over a large scale network. The middleware is an Immune-Networking architecture that provides software agents and an Immune-Net platform. Software agents are autonomous, reactive and mobile. Immune-Networking platform provides execution and support services for the software agents.

The rest of the paper is organized as follows. In section 2, we present the large scale networking challenges. Section 3 presents related works. Section 4 is an overview of the natural immune system. In section 5, we present the immune based middleware. Conclusion is given in section 6.

2. Large scale networks

The evolution of computing and networking has gone successively from mainframes, to minicomputers and later to personal computers. The rapid growth in personal computing and commodity networking has created a much larger distributed-computing environments such as computational grids[8]. These environments have the potential ability to integrate large-scale computing resources, on demand. Users as well as resources can be mobile or partially connected. User ability to compute will no longer be limited to the resources he has currently at hand or those localized statically on a set of hosts known a priori. Trends today are towards networking resources. For example, a user with a diskless machine can load a word-processor for a punctual need. He no longer needs necessarily a machine with a disk and his own word-processing software installed on it. In addition, he may not know in advance from which host he get the resource throughout his current session. The availability of the host and its resource depends on the dynamic behavior of the network (e.g., failure or network disconnection). When the user submits its request, eventually with QoS requirements, the immune middleware will process it according to the network status at that instant, and in order to satisfy the QoS constraints. The Internet is an example of a large-scale distributed characterized by rapid growth, high heterogeneity and unpredictable dynamic behavior.

A resource is an entity that provides state and functionality to be utilized by other entities. Resources are divided into two basic categories[9]: system resources and application resources. System resources are bound to specific hosts, representing hardware devises (e.g. disk) or logical system objects (e.g. socket). Application resources are software entities, possibly mobile, which are managed by an application.

A fundamental aspect to allocate a resource in a network is done via interaction between the provider of the resource and the user. We can distinguish between two resource allocation interaction mechanisms: resource allocation in the client-server paradigm and in the mobile agent based paradigm. Recently, mobile agents have been advocated as a useful mechanism to design and implement large scale network based services[15]. In the mobile agent based approach, the user initiates a request without specifying any destination server. Two scenarios can be considered[1]. In the first one, the user launches a discovery agent that jumps from site to site to seek the resource and returns back the result to the user. In the second scenario, the user launches several discovery agents(duplicated into clones) to seek the required resource. A comparison between the resource allocation performance in the client-server approach and in the mobile agent based approach is given in[1].

Benefits of Mobile agents

Mobile agents provide interesting proprieties for handling dynamic network, load balancing and disconnected operation[15]. Some of their advantages over preexisting computing paradigm such as client-server are the following[10]:

- Asynchronous autonomous interaction : agents are mostly autonomous and self-contained. They can operate asynchronously and independently.
- Dynamic adaptation : mobile agents have the ability to autonomously react to changes in their environment.
- Efficiency : the concept of an agent moving between network nodes gives it the ability to survive and to research as many resources as possible.
- Fault tolerance : the ability of mobile agents to react dynamically to adverse situations, such as sudden topology changes, makes it easier to build fault tolerant behavior, especially in a highly distributed system.
- Reduction of the network traffic : most communication protocols involve several interactions. This causes a lot of network traffic. With mobile agents, one can package up a conversation and ship it to a destination host where the interactions can take place locally[10].

3. Related works

Biological principles have been exploited in a variety of computationally based learning systems such as artificial neural networks and genetic algorithms[19]. Also, the emergence of complex collective behavior from the local interactions of simple agents is illustrated by many natural social systems, like ant colony[7] and bee colony[18]. A framework of mobile agents inspired by ant colonies has been proposed in [7] and a framework of mobile agents inspired by bee colonies has been proposed in[18]. Immunological principles and functionalities from computational viewpoint have been applied to computer security problems[12, 17]. In particular, the mechanisms to extract unique signatures from antigens and ability to recognize and classify dangerous antigens[6]. In[13], intelligent agents inspired by the Human immune system functionalities have been proposed to provide support for parallel processing. In our work, a mobile agent based middleware inspired by the immune system is proposed to deal with resource allocation problem in large scale networks.

In the following section, we briefly describe the functionalities and the principle characteristics of the immune system[3, 11, 13, 14, 17].

4. Immune system overview

The immune system defends the body against harmful diseases and infections. It is capable of recognizing most pathogens and eliminating them from the body. To do this, it must perform patter recognition tasks to distinguish molecules and cells of the body(self) from foreign ones(non self). Once pathogens have been entered the body, they are handled by the innate immune system and by the adaptive immune response. The innate and adaptive immune response are produced primarily by leukocytes. Among the several different types of leukocytes, there are phagocyte and lymphocytes. The phagocytes are the first lines of defense for the innate immune system. They engulf the antigen and destroy them. The adaptive immune system consists primarily of lymphocytes that circulate through the body in the blood and lymph system. There are two categories of lymphocytes, the B-cells and T-cells. The B-cells are developed in the bone marrow and the T-cells are developed in the thymus. The T-cells are divided into two major cells: the T-helper cells and the T-killer/suppressor cells. The principle function of T-helper cells is to potentiate immune response by the secretion of specialized factors that activate other cells to fight off infection. The T-killer/suppressor are important in directly killing an antigen. The major function of B-cell is the production of antibodies in response to foreign antigen. If the antibodies bind to antigens it acts a signal for phagocytes or T-killer cells to engulf and kill them. An other class of soluble mediators named cytokines, realized by T-helper, cells to communicate with other cells in the innate and adaptive immune system. In the innate system, phagocytes take up antigens and present them to the T-helper cells for antigen recognition. If the T-helper recognizes an antigen, it will release cytokines to activate the phagocytes to destroy it. T-helper cells also communicate with B-cells to help them recognize and destroy the antigen. The immune system maintains a memory of new encountered antigen. Then, when the body is reinfected with the same antigen, the secondary response of the immune system, based on the immunological memory, is very specific and rapid.

Principal characteristics

The immune system has a useful set of organizing principles. Some of them are described in the following[17]:

- Multi-layered and diversity : The immune system is composed of different layers of defense: physical, physiological, innate and adaptive. The physical barrier is the skin. The physiological barrier provides inappropriate living conditions for foreign organisms such as pH and temperature. For pathogens that elude these two basic defense mechanisms, there are innate and adaptive (i.e., acquired) immune response.
- self-organization and distributivity : Lymphocyte in the immune system are able to determine locally the presence of an infection. No central coordination takes place. The immune system provides a good example of self organization and highly distributed architecture.
- Disposability : No single component of the immune system is essential, that is, any cell can be replaced.
- Autonomy : The immune system does not require outside management; it autonomously classifies and eliminates pathogens.
- Adaptability : The immune system learns to detect new pathogens, and retains the ability to recognize previously seen pathogens through immune memory. Our middleware exploits these adaptable functionalities: learning to recognize new resources and remembering the signatures of previous allocation.
- Self-regulatory : the human immune system replicate detectors to deal with replicate pathogens. Otherwise, the pathogens would quickly overwhelm any defense. Our agents-based system are subject to a similar numbers game, by replicating the agents into clones in order to propagate the resource information and allocation requests.

5. Immune-Networking middleware

Functionalities that the immune system exhibits are mapped to our immune based middleware. The middleware architecture consists of an Immune-Net platform and mobile software agents that correspond to Lymphocyte cells.

5.1. Immune-Net platform

Immune-Networking platform provides the execution environment and support services for the mobile software agents. These services are the following:

- Communication service: this service allows the mobile agents to communicate with each other.
- Migration service: this service provides migration statement to carry out the proactive mobility of agents.
- Resource sensing service: this service detects type and available resources over the network.
- Request sensing service: this service handles user requests. This service corresponds to phagocytes in the immune system.

- Resource allocation service: this service allocates physical and logical resources to agents.
- Life cycle service: this service manages the mobile agents life-cycle. It provides operations to initialize, activate, suppress, clone and reproduce mobile agents.
- Security service : this service guarantees confidentiality, integrity, and authentification of information a mobile agent holds. It is also responsible for access control to the resources.

5.2. Mobile software agents

Analogous to lymphocytes in the immune system, software agents are autonomous, adaptive, reactive and mobile. The important component of our research is to analyze the computational aspects of the immune system and integrate them in a multi-agent system. Mobility gives agents the ability to roam around machines and routers to explore and monitor resources and user requests. They can learn and adapt to their environment and can mutually recognize new resources and requests. They can collaborate to produce specific responses. Several types of agents are considered in our middleware. The role of each agent type is unique but they can work in collaboration.

- B-monitoring agents: They correspond to B-cells in the human immune system. The role of these agents is very important, they roam around the network with specific functions. Some of these agents may work for monitoring resource changes, while others transport the knowledge about recognized resources distributed in the network.
- Communicator agents: they serve as negotiators in order to elaborate liaison among agents. They correspond to cytokines secreted by T-cells to activate B-cells in the natural immune system.
- T-helper agents: activated by request sensing service through communicator agent. The request sensing service (i.e., phagocytes) takes up a request and presents it to the T-helper agent for resource recognition. If the T-helper agent can check the required resource availability, it activates the resources sensing service to satisfy the request. Otherwise, The T-helper agent alerts B-monitoring agents.
- T-suppressor agents: they may be activated by T-helper agents or by request sensing service through communicator agents. These agents correspond to T-suppressor agent in the human immune system.

As in the natural immune system, the middleware has the following operating modes on each network node:

- Monitoring mode: when resource sensing service monitor the resources (i.e, updating, deleting and dissemination).
- Recognition mode: the system is in the recognition mode, on a particular node, when request sensing service receives user requests or resource notification and takes appropriate decisions (e.g., communicator agent and T-helper agent activation, ...).
- Response mode: the system takes appropriate action by activate specific type of agents such as, T-suppressor agent, T-helper agent, etc.

The development of this immune-based middleware is an ongoing effort. Future work will address the performance issues.

6. Conclusion

In this paper, we have presented an autonomous decentralized system based on mobile agents and inspired by the immune system as an alternative to the traditional client-server paradigm. The proposed immune based middleware integrates several functionalities of the natural immune system. The middleware exhibits self-organization with support for mobility, scalability, and adaptability to user and network conditions changes. It brings answers to some large scale networking challenges.

References

[1] M. Bakhouya, J. Gaber, and A. Koukam. Observations on client-server and mobile agent based paradigms for resource allocation. *Workshop on Performance Modeling, Evaluation, and Optimization of Parallel and Distributed Systems PMEO-PDS02 (IPDPS 2002 Workshop), Fort Lauderdale, Florida*, April 15-19, 2002.

[2] A. Bieszczad, B. Pagurek, and T. White. Mobile agents for network management. *IEEE Communications Surveys*, 1(1):1, 1998.

[3] R. G. Cardenas. Distributed systems and computer security. *Studia Informatica Universalis, OPODIS 2000 special issue*, 2:235–243, 2000.

[4] D. Chess, C. Harrison, and A. Kershenbaum. Mobile agents: Are they a good idea? *IBM T. J. Watson Research Center*, 1994.

[5] J. Dale. A mobile agent architecture to support distributed resource information management, phd thesis. *http://www.mmrg.ecs.soton.ac.uk*, 1997.

[6] D. Dasgupta. Immunity-based intrusion detection system: A general framework. *In the proceedings of the 22nd National Information Systems Security Conference (NISSC)*, 1999.

[7] S. Fenet and S. Hassas. A.N.T.: a distributed network control framework based on mobile agents. *In Proceeding of the International ICSC Congress on Intelligent Systems And Applications, CI'2000 (in ISA'2000)*, 2000.

[8] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid, enabling scalable virtual organizations. *Intl J. Supercomputer Applications*, 2001.

[9] Y. Gidron, O. H. I. Ben-Shaul, and Y. Aridor. Dynamic configuration of access control for mobile components in fargo. *Concurency and Computation*, 13(1):5–21, January 2001.

[10] S. Green, B. Nangle, P. cunningham, F. Somers, and R. Evans. Software agents: A review. *http://citeseer.nj.nec.com/green97software.html*, 1997.

[11] S. A. Hofmeyr. An immunological model of distributed detection and its application to computer security. *PhD thesis, University of New Mexico*, 1999.

[12] S. A. Hofmeyr and S. Forrest. Architecture for an artificial immune system. *Evolutionary Computation*, 8(4):443–473, 2000.

[13] R. L. King, A. B. Lambert, S. H. Russ, and D. S. Reese. The biological basis of the immune system as a model for intelligent agents. *Second Workshop on Bio-Inspired Solutions to Parallel Processing Problems, Lecture Notes In Computer Science 1586, pp. 156-164*, 1999.

[14] P. A. Linnemeyer. The immune system–an overview. *http://www.thebody.com/step/immune.html*, 1993.

[15] P. Quinnec, M. Filali, P. Mauran, and G. Padiou. Describing mobile computations with path vectors. *Studia Informatica Universalis, OPODIS 2000 special issue*, 2:221–234, 2000.

[16] A. Sahai. Conception et réalisation d'un gestionnaire mobile de réseaux fondé sur la technologie d'agents mobiles. *ftp://ftp.irisa.fr/techreports/theses/1999/sahai.ps.gz*, 1999.

[17] A. Somayaji, S. Hofmeyr, and S. Forrest. Principles of a computer immune system. *In Proceedings of the Second New Security Paradigms Workshop*, pages 75–82, 1997.

[18] M. Wang and T. Suda. The bio-networking architecture. *http://netresearch.ics.uci.edu/bionet/publications/bionet-TR00-03.doc*, 2000.

[19] A. Zomaya, F. Ercal, and S. Olariu, editors. *Solutions to Parallel and Distributed Computing Problems: Lessons from Biological Sciences*. Jhon Wiley, 2000.