Dynamic Load Balancing of the Bio-network Based on Bio-Entity Migration

Xiangfeng Zhang[1] **Yongsheng Ding**[2]

[1]College of Electric, Shanghai Dianji University, Shanghai 200240, P. R. China
[2]College of Information Sciences and Technology, Donghua University, Shanghai 201620, P. R. China

Abstract

Inspired by the principles and concepts of neuroendocrine-immune systems, the bio-network architecture has been proposed. The migration behavior of bio-entities and the migration stability are analyzed. A dynamic load balancing algorithm is proposed based on bio-entities migration behavior. According to the decreasing tasks in the bio-network, the simulation experiments are presented to compare the average response time per task through adopting or not adopting the algorithm. The results demonstrate that the algorithm is a feasible scheme for dynamic resource allocation.

Keywords: Dynamic load balancing, Bio-entities migration, Bio-network architecture, Neuroendocrine-immune system

1. Introduction

In biological environments, large-scale systems have developed many mechanisms which can adapt themselves to the environments. The biological inspired information systems have recently received much attention, and provided many effective intelligent technologies and methods for engineering fields. In the past several decades, artificial neural network (ANN) and artificial immune system (AIS) were studied widely. Currently researchers have attended importance to the interrelationship among the neural, endocrine, and immune systems in the fields of physic, biology and intelligent computation [1]-[3].

With the development of computer and network technology, there are many advanced distributed requirements on future Internet. Future Internet should be capable of scalability, extensibility, maintainability, and self-evolution [4]. Considering the complex characteristics of biological network and the features of future Internet, we have abstracted computing framework inspired from the principles and mechanisms of neuroendocrine-immune (NEI) systems. The NEI-based bio-network architecture has also been advanced [3]. The network performance should be enhanced with the increment of usable network service in the bio-network. In other words, the bio-network should deal with more tasks through providing more service during the same period. Thus, the bio-network can maintain load balancing [5].

Dynamic load balancing is an important issue to be solved in order to achieve the above characteristic. Maintaining network load balancing is one of the important factors of optimizing network resources. The dynamic load balancing of loosely synchronous and computationally intense adaptive applications has been studied extensively in the past two decades. Distributed applications employ load balancing to improve the scalability of the system in various ways and at various levels [6]. However, it is important to transmit tasks from the overload node (such as computer, workstation) to the lighted-load one. We discuss dynamic load balancing of the bio-network inspired from the homeostasis and integrated stability of NEI systems.

The rest of this paper is organized as follows: Section 2 discusses the bio-network architecture. Section 3 discusses bio-entities migration behavior, analyzes migration stability, and proposes a dynamic load balancing algorithm based on the bio-entities migration. Section 4 then analyzes the results of our experiments. Section 5 concludes with some future plans.

2. The bio-network architecture

The nervous, endocrine and immune systems are three important physiological modulation systems in a body. There are not only large loops among them, but also bidirectional communication between them to maintain normal actions [7]. Theses interactions among the systems are classified as long loop, short loop and super-short loop.The function and structure of a body will change in some degree when the environments change. Thus the body maintains dynamic balancing with its environments, which is called adaptation. The result of longtime adaptation is evolution. One system in the body is responsible for several physiological functions. However, the outer and inner environments of the body are variational. Hence, a set of accurate modulating organizations adjusts continuously the organs and systems, harmonize and cooperate to develop a uniform body. They adjust physiological functions to compose a solid network structure and collaborate to answer for adaptive actions of different environments.

From the biological foundations of NEI system, we have abstracted some important principles and mechanisms (such as individual interaction, radiation, aggregation, and adaptation), which satisfy key requirements of future network. Integrating them into a uniform framework, we have designed the bio-network architecture and its simulation platform by agent-oriented approaches [3].

The bio-network architecture consists of the bio-network platform, bio-entities and their survivable environment. The bio-network platform is a software framework fully implemented by Java language. It includes bio-network core service layber and bio-network low-level functional modules established in a network node. The bio-network core service layer provides a set of general-purpose runtime services that are frequently used by bio-entities, such as lifecycle, bio-entities migration, evolution state management, interaction control, and security authentication. The low-level functional modules manage low-level network and system resources and allocate resources to a bio-entity.

A bio-entity is a basic component and an autonomous agent and analogous to a cell (such as immune cell, neuron) of NEI system. It represents a user request or a simple service. Several bio-entities interact and develop complex network service and network applications. The bio-entity consists of attributes, service information, behaviors, and communication mechanisms. Its behaviors include mainly migration, replication, reproduction, interaction, and announcement, etc.

3. Dynamic load balancing algorithm based on bio-entities migration

3.1. Bio-entities migration behavior

The bio-network platform runs in an asynchronous distributed system on a network node, as shown in Fig. 1. Thus an overlay network environment is formed with wide-area bio-network platforms [8]. Several nodes can form a community niche and the niche is a logically defined area where the bio-entities in a community can learn from their surrounding environment. A bio-entity can sense which bio-entities are in the community niche, what services they perform, and which resources it can access to. This helps it create a new application or join an existing community. Physical proximity among network nodes is used to define a community niche in this study.

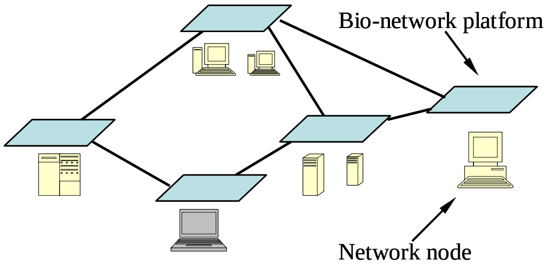

Fig. 1: The network environment based on bio-network platform.

The bio-network platform provides the bio-entity migration service which supports the migration behavior of the bio-entities. The migration behavior involves determining where and when to migrate through considering the cost/benefit credit units of migrating towards an adjacent community niche. The migration will be used to find useful partner bio-entities and acquire new relationships. There are three methods to migrate: near migration in a community niche, far migration in a community niche, and migration between two community niches.

3.2. The stability analysis of migration

Good bio-entities replicate or reproduce their child bio-entities to execute user requests effectively, while bad bio-entities die because of the old age or being lack of credit units. A bio-entity can exist longer and even produces a new bio-entity to provide services if it has enough credit units. The user requests in a community niche are relatively unvarying and the bio-entities in the same community niche have the same birth rate and death rate. Assume that the birth rate and the death rate in niche i is b_i and d_i, respectively. So the natural increase rate c_i is $c_i = b_i - d_i$ ($i = 1,2,...,n$) in niche i. The bio-entities keep active and c_i is greater than zero if the user requests are sufficient. These bio-entities in different community niches have different natural increase rates because of different user requests. If a bio-entity migrates randomly and hence consumes many credit units, it dies and cannot continue to provide services for the request. Next, we discuss the stability of the bio-entity migration.

Suppose the dynamic model of the interaction among p communities,

$$dx_i / dt = f_i(x_1,..., x_p) \quad i = 1,..., p \tag{1}$$

As to an arbitrary subsystem,

$$dx_i^k / dt = f_i(x_1^k,...x_p^k) + m_{ik}(x_1^k,..., x_p^k)$$
$$k = 1,..., n\ ,\quad i = 1,...p \tag{2}$$

where,

k is the subsystem number,

f_i is the interaction structure of individuals in subsystem k,

m_{ik} describes migration effects acted by all other subsystems on the subsystem k.

Here we just discuss the simple situation. Suppose that the number of user requests in one community niche is stable. There are some service bio-entities in another community niche. The number of the user requests and that of service bio-entities are $x(t)$ and $y(t)$ at the time *t*, respectively. The general dynamics model of their interaction is

$$\begin{cases} dx / dt = f(x, y) \\ dy / dt = g(x, y) \end{cases} \tag{3}$$

To simplify the model, m_{12} and m_{21} are introduced as follows,

$$(m_{12})_x = r + \rho$$
$$(m_{21})_x = r - \rho$$
$$(m_{12})_y = s + \sigma$$
$$(m_{21})_y = s - \sigma$$

where, m_{ks} ($k, s = 1,2$) is larger than or equal to zero and it is a migration strength factor, which means the independency of migration route. The parameter ρ and σ are used to distinguish a contrary migration strength. Suppose

$$(m_{12})_x > (m_{21})_x$$
$$(m_{12})_y > (m_{21})_y .$$

Then, ρ and σ satisfies with the inequation $0 < \rho < r$ and $0 < \sigma < s$, respectively.

The dynamics model equation group of the combination community is

$$\begin{cases} \dot{x}^1 = f(x^1, y^1) - (r - \rho)x^1 + (r + \rho)x^2 \\ \dot{y}^1 = g(x^1, y^1) - (s - \sigma)y^1 + (s + \sigma)y^2 \\ \dot{x}^2 = f(x^2, y^2) + (r - \rho)x^2 - (r + \rho)x^2 \\ \dot{y}^2 = g(x^2, y^2) + (s - \sigma)y^1 - (s + \sigma)y^2 \end{cases} \tag{4}$$

where the superscript is the niche's serial number. More than one bio-entity migrates and the strength direction is different, which can be expressed as follows,

$$r + s > 0\ ,\quad \rho^2 + \sigma^2 \neq 0 \tag{5}$$

If the system (4) has a non-common balanceable state,

$$z\wedge = (\hat{x}^1, \hat{y}^1, \hat{x}2, \hat{y}^2) .$$

The structure matrix of the balance point is

$$\mathrm{A}(z\wedge) = \begin{pmatrix} A(\hat{x}^1, \hat{y}^1) - diag\{r - \rho, s - \sigma\} & diag\{r + \rho, s + \sigma\} \\ diag\{r - \rho, s - \sigma\} & A(\hat{x}^2, \hat{y}^2) - diag\{r + \rho, s + \sigma \end{pmatrix} \tag{6}$$

Where, $A = (\frac{\partial f_i}{\partial x_i}|_{\hat{x}})$ .The equation (4) can be expressed with the equation (7) if the bio-entities migrate to provide service,

$$\begin{cases} \dot{x}^1 = \alpha x^1 - \beta x^1 y^1 \\ \dot{y}^1 = -\gamma y^1 + \delta x^1 y^1 - (s - \sigma)y^1 + (s + \sigma)y^2 \\ \dot{x}^2 = \alpha x^2 - \beta x^2 y^2 \\ \dot{y}^2 = -\gamma y^2 + \delta x^2 y^2 + (s - \sigma)y^1 - (s + \sigma)y^2 \end{cases} \tag{7}$$

There is a positive balanced point if γ is large than 2σ,

$$\hat{y}^{1,2} = \hat{y} = \alpha / \beta$$
$$\hat{x}^{1,2} = \hat{x} \pm \Delta x = (\gamma \mp 2\sigma) / \delta \tag{8}$$

We can see that $\hat{x}^1$ is less than $\hat{x}^2$ from the equation (8), which shows that the equation (8) is gradual stabilization. And this shows that the migration behavior can keep community niches in a stable state.

3.3. Dynamics load balancing algorithm

Load sharing and task migration are the issues which have been widely researched in dynamic load balancing algorithms [9]. When newly created tasks arrive randomly into the network, nodes can become heavily loaded while others are idle or lightly loaded. The main objective of load sharing is to develop task assignment algorithms to migrate tasks from heavily to lightly loaded nodes. And no nodes are idle when other tasks waits to be processed. A dynamic load balancing algorithm consists of four major components: load measurement, task transmission rule, selection scheme, and tasks incept.

(1) Load Measurement.

Load measurement decides which system information will be used. Five parameters are adopted to describe load information and they are CPU utilization ratio, memory, disk I/O access ratio, response time and the total number of course.

(2) Task Transmission rule.

Threshold scheme is used to decide whether a task is transmitted to another node. If a node accepts tasks more than those it can deal with, it will transit the newly receiving task to others.

Suppose that there are *N* nodes in the bio-network. $W(N_i)$ is the weight of the node N_i, $T(N_i)$ is the number of tasks assigned to the node N_i, $\sum T(N_i)$ and $\sum W(N_i)$ are the total tasks and the total weights of those nodes during a period of time, respectively. Define

$$T(N_i)/\sum T(N_i) = W(N_i)/\sum W(N_i) \qquad (9)$$

Equation (9) means that the tasks are assigned according to the ratio of the weight of each node to the total ones. The dynamic weight is computed in terms of the parameters. And it reflects correctly the status of loads in the node and predicting the potential change of loads. We set a constant coefficient α_j for each parameter in order to adjust each parameter's proportion and all the coefficients satisfy the equation $\sum \alpha_j = 1$. For any node N_i, the dynamic weight is computed as follows:

$$W_D(N_i) = \sum_{j=1}^{m} \alpha_j Load_j(N_i) \qquad (10)$$

Where, $Load_j(N_i)$ is the load value of the *j*th parameter of the node N_i.

We modify the weight to make it more reasonable and the final weight is computed using the equation (11),

$$W_F(N_i) = (1-\beta)W_I(N_i) + \beta W_D(N_i) \qquad (11)$$

Where, β is the coefficient and $W_F(N_i)$ and $W_I(N_i)$ are the final weight and the initial weight of node N_i, respectively.

If $W_D(N_i)$ is equal to $W_I(N_i)$, the weight does not change and the load is to the limit. If $W_D(N_i)$ is larger than $W_I(N_i)$, the final weight becomes large and the node is light-loaded. The node can execute more tasks. Or else, the final weight becomes small and the node is overloaded. The node should allocate tasks to others.

(3) Selection Scheme.

The node schedules locally those tasks that are little relative with others. For instance, a request bio-entity migrates to the node N_i and sends request service. If the node N_i is overloaded, it will select other neighbor nodes according to their final weights. Then it sends the irrelative bio-entities to other node that can provide service.

(4) Tasks Acceptance.

Tasks acceptance decides which node the tasks are sent to. It is relevant to the execution performance and workload of the node, and the communication spending among nodes. At last, each node stores the results and updates its tasks.

The load balancing scheme is shown in Fig. 2.

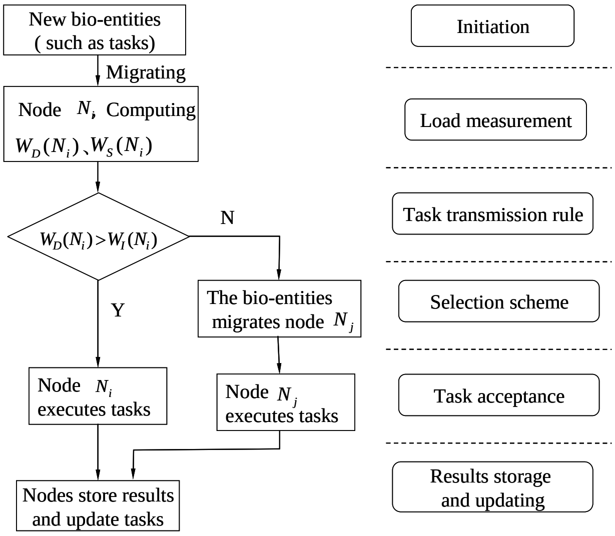

Fig. 2: The load balancing scheme among nodes.

4. Simulation and discussion

When users access network information, they require customized services which the servers cannot provide in time. It is necessary to adjust services of Web servers with load balancing approach so that the tasks in the network can be processed efficiently. The simulation experiment is made on our designed simulation platform constructed on Windows 2000 operation system with Intel Pentium 4 processor (2.4 GHz) and 512 MB RAM. A simulated network is a 32×24 topology one with 768 network nodes. Each node in the network is a bio-network platform. The tasks and services represented by bio-entities are deployed on the platform.

The performance of the proposed method is compared to a random allocation scheme. Suppose that the constant coefficient α_i is (0.15, 0.05, 0.3, 0.2, 0.3), which is CPU utilization ratio, disk I/O access ratio, memory, the total number of course and response time, respectively. Memory and response time are more important than others. The simulation evaluates the performance of the proposed method using the response time per task, as shown in Fig. 3.

Given that the tasks decrease in the bio-network, we can see that the response time per task decreases rapidly if the load balancing algorithm is adopted. However, the response time per task changes inconspicuously using a random method. The results show that the bio-entities can migrate efficiently nodes and easily achieve services under the load balancing scheme.

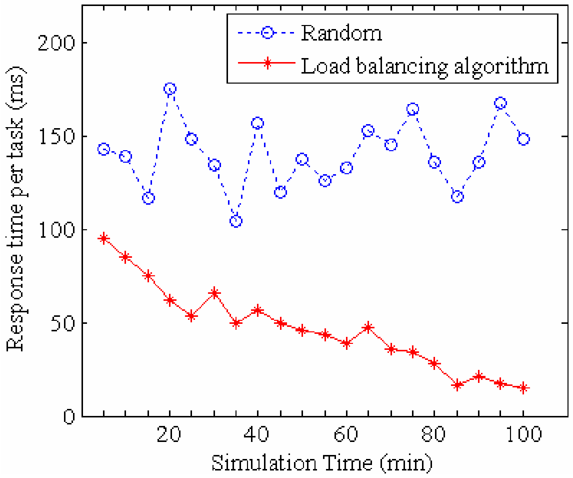

Fig. 3: The response time per task.

5. Conclusions and future work

We have originally applied the principles and concepts of neuroendocrine-immune systems to the design of bio-network architecture. The migration behavior of bio-entities and migration stability are analyzed. The migration behavior keeps the bio-network stable. A dynamic load balancing algorithm is proposed based on the bio-entities migration behavior. Finally, given that the tasks decrease in the bio-network, we simulate the response time per task on the simulation platform. Experiment results show that the response time decrease quickly using the proposed algorithm. It provides a feasible scheme for dynamic resource allocation.

The next work is on issue about improving other bio-entities behaviors. Usually, the amount of tasks and the size of ontology networks are huge. Therefore, it is necessary to evolve bio-entities. In addition, more experiments will be designed to evaluate the load balancing algorithm in different network environments.

Acknowledgment

This work is partially supported by Program for New Century Excellent Talents in University from Ministry of Education of China (No. NCET-04-415), the Cultivation Fund of the Key Scientific and Technical Innovation Project from Ministry of Education of China(No. 706024), International Science Cooperation Foundation of Shanghai (No. 061307041).

References

[1] C.L. Raison, J.F. Gumnick, and A.H. Miller, Neuroendocrine-immune interactions: Implications for health and behavior. In D. Pfaff, A. Arnold, A. Etgen, S. Fahrbach, and R. Rubin, editors, *Hormones, Brain and Behavior*, Academic Press, San Diego, 2002.

[2] P. Vargas, R. Moioli, and L.N. De Castro, Artificial homeostasis: A novel approach. *Proceedings of European Conference on Artificial Life*, pp. 754-763, 2005.

[3] X.F. Zhang, L.H. Ren, and Y.S. Ding, A neuroendocrine-immune system inspired bio-network architecture and its simulation platform for complex network applications. *Dynamics of Continuous, Discrete and Impulsive Systems - Series B: Applications & Algorithms*, 13: 21-25, 2006.

[4] E. Putrycz and G. Bernard, Connecting frameworks: Case study with middleware-based load balancing. *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops*, pp. 126-131, 2003.

[5] Z. Lan and V.E. Taylor, Dynamic load balancing of SAMR applications on distributed system. *Proceedings of the 2001 ACM/IEEE Conference on Supercomputing*, pp. 36-48, 2001.

[6] K. Barker, A. Chernikov, N. Chrisochoides, and K. Pingali, A load balancing framework for adaptive and asynchronous applications. *IEEE Transactions on Distributed Systems*, 15: 183-192, 2004.

[7] H. O. Besendovsky and A. del Ray, Immune-neuro-endocrine interactions: Facts and hypotheses. *Nature*, 249: 356-358, 1996.

[8] X.Y. Zhang, Q. Zhang, Z.S. Zhang, G. Song, and W.W. Zhu, A construction of locality-aware overlay network: Overlay and its performance. *IEEE Journal on Selected Areas in Communications*, 22: 18-28, 2004.

[9] A.Y. Zomaya, Observations on using genetic algorithms for dynamic load-balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12: 899-911, 2001.